Scientific Research

Positioning

ISSN: 2150-850X (Print), 2150-8526 (Online)

Vol. 8 No. 1 June 2010

www.scirp.org/journal/positioning

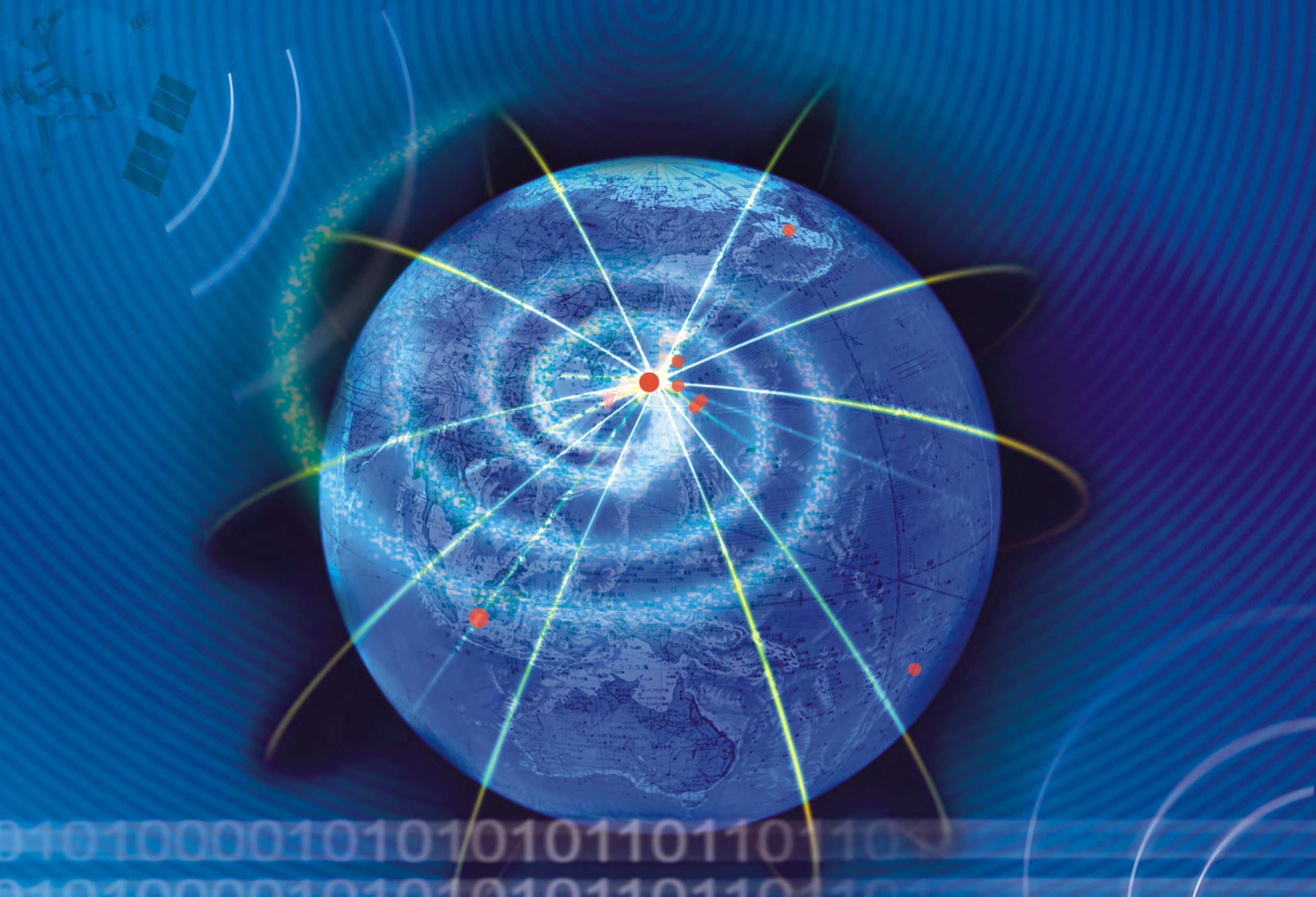


Table of Contents

Positioning (POS)

Journal Information

SUBSCRIPTIONS

The *Positioning* (Online at Scientific Research Publishing, www.SciRP.org) is published quarterly by Scientific Research Publishing, Inc., USA.

Subscription rates:

Print: $50 per issue.
To subscribe, please contact Journals Subscriptions Department, E-mail: sub@scirp.org

SERVICES

Advertisements
Advertisement Sales Department, E-mail: service@scirp.org

Reprints (minimum quantity 100 copies)
Reprints Co-ordinator, Scientific Research Publishing, Inc., USA.
E-mail: sub@scirp.org

COPYRIGHT

Statements and opinions expressed in the articles and communications are those of the individual contributors and not the statements and opinion of Scientific Research Publishing, Inc. We assumes no responsibility or liability for any damage or injury to persons or property arising out of the use of any materials, instructions, methods or ideas contained herein. We expressly disclaim any implied warranties of merchantability or fitness for a particular purpose. If expert assistance is required, the services of a competent professional person should be sought.

PRODUCTION INFORMATION

For manuscripts that have been accepted for publication, please contact:
E-mail: pos@scirp.org

Requirements for the next generation standardized location technology protocol for location-based services

Lauri Wirola, Ismo Halivaara and Jari Syrjärinne
Nokia Inc., Finland

Abstract

The booming location-based services business requires more accuracy and availability from positioning technologies. While several proprietary location and positioning protocols have been developing in the market, scalable and cost-effective solutions can only be realized using standardized solutions.

Currently the positioning protocol standardization is concentrated in the 3GPP and 3GPP2 that define Control Plane (CP) positioning technologies for Radio Access Networks' native use. The limitations of the control plane in terms of architecture and bearer protocols are necessarily reflected in the CP positioning protocols and limit the feature sets offered. In addition to 3GPP/2 positioning technologies are also defined in WiMAX Forum and in IEEE for WLAN networks.

Location protocols in IP-networks, such as OMA SUPL (Open Mobile Alliance Secure User Plane Location protocol), encapsulate the CP positioning protocols. Thus the limitations of the CP protocols have also been copied to the User Plane, although the bearer there would be much more capable.

Due to the shortcomings in the CP positioning protocols, standardization activity for a new bearer-independent positioning protocol is proposed in order to fulfil the needs of the future location-based services. This paper discusses the current solutions, trends in the location technologies and outlines requirements for the future location technology protocol in terms of protocol features and data content.

The development of a generic positioning technology protocol is seen as an important development towards a convergence in the location protocols and the capability to provide location-based services irrespective of the bearer network. This has a major impact on the service development as well as user experience.

Keywords: User Plane, Positioning protocol, Assisted GNSS, Fingerprint, Hybrid positioning

1. Introduction

Developing positioning and location standards has substantial market demand. Already now AGPS-enabled (Assisted GPS) mobile terminals constitute a significant share of the global navigation device market. The 2008 annual GPS-enabled smart phone sales are estimated above 30 million units and the analysts estimate that in 2011 the annual sales surpass 90 million units (Canalys, 2008). Moreover, modern smart phones are location-aware at least through the cellular network base station information. Finally, laptops can be made location-aware using WLAN-based positioning methods.

Positioning protocol standardization is concentrated in 3GPP (The Third Generation Partnership Project) and 3GPP2, which define positioning protocols for the Control Planes of GERAN (GSM EDGE Radio Access Network), UTRAN (UMTS Terrestrial RAN), E-UTRAN (Enhanced UTRAN) and CDMA (Code Division Multiple Access) networks. GERAN, where EDGE stands for Enhanced Data rate for Global Evolution, is better known as GSM (Global System for Mobile communications). UTRAN, where UMTS stands for Universal Mobile Telecommunications System, is commonly referred to as WCDMA (Wide-band CDMA). Finally, E-UTRAN is also known as LTE (Long-Term Evolution).

The Release 8 of GERAN standard will include the possibility to provide terminals with assistance data for all the existing and some future GNSSs (Global Navigation Satellite System). The assistance includes, among other things, the navigation model (orbit and clock parameters), reference location and reference time. In an assisted situation, the receiver does not need to download the navigation model from the satellites, but receives it over the cellular network to considerably reduce the time-to-first-fix. Moreover, location and time data improve sensitivity significantly. The positioning is thus enabled in adverse signal conditions such as urban canyons. The improvement in user experience is significant compared to the performance of the autonomous GPS or simple cell-ID based positioning.

In addition to the RAN-independent AGNSS data each 3GPP location protocol also contains RAN-specific items. For instance, RRLP (3GPP-TS-44.301) (Radio Resource LCS Protocol, LCS LoCation Services) for GERAN networks and RRC (3GPP-TS-25.031) (Radio Resource Control protocol) for UTRAN networks include time difference and round trip time measurements, respectively, allowing for native RAN-based network positioning. Moreover, the reference time is given in a RAN-specific way by binding the cellular frame timing to the GNSS time.

Solutions for IP-networks include OMA (Open Mobile Alliance) SUPL (Secure User Plane Location protocol) Release 1 (OMA-TS-SUPL-1-0, 2007) and (draft) Release 2 (OMA-TS-SUPL-2-0, 2009) that encapsulate Control Plane positioning protocols defined by 3GPP/3GPP2 as sub-protocols to ULP (User plane Location Protocol). In addition to the capabilities of 3GPP and 3GPP2 positioning protocols the (draft) SUPL Release 2 adds items for, for instance, WLAN- (Wireless Local Area Network, IEEE 802.11) and WiMAX-based (Worldwide Interoperability for Microwave Access) positioning.

Currently the AGNSS-based positioning methods are essentially the only standardized positioning solutions available for global LBS (Location-Based Services). The native RAN-based methods are not widely deployed and their accuracy is varying. Moreover, the WLAN-based positioning capability in the (draft) OMA SUPL Release 2 is quite limited.

The future location services require more accuracy and availability from the standardized positioning solutions, because the use cases and user appetite for LBS will become more demanding. The requirement for increased availability can be understood by noting that people tend to stay indoors majority of the time, which is also the environment, where AGNSS performance is the worst. Accuracy requirement is, naturally, a question of application – location-sharing in a social network may require only cell-based positioning. On the other hand guiding a person to the correct entrance instead of just address requires higher accuracy than the current standardized solutions can offer.

However, when it comes to introducing new features into the standards, the problems lie in the currently utilized Control Plane positioning standards being bound to the architecture and protocol limitations in the respective RANs. Moreover, because those protocols evolve from RAN needs (mainly emergency call positioning requirements), Control Plane positioning protocols will not add support, for example, for novel signal-of-opportunity -based positioning technologies. Therefore, a new standardized location protocol is required to introduce and implement new novel features and positioning technologies.

This article reviews the existing positioning protocols, discusses the future location needs and shows the limitations of the current solutions. Finally, based on the future needs and use cases, requirements are outlined for the new standardized location technology protocol that is flexible, scalable and comprehensive. In the long term the sought goal is the convergence towards a single generic User Plane location technology protocol.

It should be emphasized that in this article the term location technology protocol refers strictly to protocols associated with obtaining the position estimate of the user using different location technologies. The services including sharing location with third parties, security and privacy are out-of-scope of this article. The same also applies to the term location technology which refers to technologies related to obtaining the plain position information.

2. Radio Positioning Protocols in Different Radio Access Networks

3GPP TS 44.031

Radio Resource LCS Protocol (RRLP) 3GPP TS 44.031 (3GPP-TS-44.031) for the Control Plane of GERAN networks is defined in the 3GPP GERAN (General Radio Access Network) Working Group 2. RRLP is a stand-alone positioning protocol used in the communication between the Mobile Station (MS) and the SMLC (Serving Mobile Location Centre). RRLP carries information on the positioning methods, such as MS-assisted and MS-based modes, as well as assistance data.

RRLP also enables reporting Enhanced Observed Time Difference (EOTD) measurements as well as delivering information about the cell tower locations and real-time differences (RTD) between the base stations to the MS for MS-based EOTD that can be used as an alternative to or in combination with Assisted GNSS (AGNSS). EOTD is based on trilateration of the MS with respect to the base stations. However, EOTD requires relatively expensive infrastructure investments in the network (LMUs, Location Measurement Unit for measuring the RTDs) and, hence, its deployment has been very limited.

The release 98 of RRLP defined the support for Assisted GPS and EOTD. The Release 7 of the RRLP brought in the support for A-Galileo and for multi-frequency measurements (including carrier-phase measurements), but also a generic structure for easy addition of other satellite systems. Finally, the Release 8 adds the support for GLONASS (Global Navigation Satellite System), QZSS (Quasi-Zenith Satellite System), Modernized GPS as well as various SBAS (Space-Based Augmentation System), such as WAAS (Wide-Area Augmentation Service) and EGNOS (European Geostationary Navigation Overlay Service).

Fig. 1 shows a simplified functional LCS architecture (3GPP-TS-43.059) in the GERAN network. The functional components in addition to MS, SMLC and LMU are BSC (Base Station Controller), BTS (Base Transceiver Station), CBC (Cell Broadcast Centre) and BSS (Base Station System). Note that the GERAN network consists of several BSS entities. The location requests originating from location clients are directed to the SMLC that handles the requests.

In the example of Fig. 1 the SMLC and CBC are integrated in BSC, although they can also be standalone components. From the positioning point of view the role of SMLC is to be the termination point of RRLP and CBC is responsible for broadcasting the assistance data to all the MSs within the cell (this is an alternative channel to distributing data over RRLP).

Moreover, an LMU can be, for instance, an entity with a GPS-receiver measuring the cellular time – GPS time relation for assistance data purposes. As mentioned, LMU is also needed for EOTD for measuring time relations between base stations. The LMU can also be a separate entity from BTS.

Although the GERAN LCS architecture is not the primary focus of this paper, the example given works to show that the Control Plane positioning protocols are strictly bound to the underlying architecture, which linkage is necessarily reflected also in the RRLP.

The RRLP includes two primary choices for the location of the position determination. In an MS-based mode the terminal autonomously determines its position taking advantage of the assistance that the terminal receives from the network. In contrast, in an MS-assisted mode the terminal typically receives only a measurement request and minimal assistance for fast signal acquisition, such as code phase search window in the AGNSS case, from the network and reports measurements to the SMLC, which determines the position. The measurements may either include GNSS measurements or EOTD measurements or alternatively both types for a hybrid solution. One to three sets of measurements can be requested and delivered to the SMLC.

In addition to MS-based and MS-assisted modes the range of methods in RRLP also includes MS-assisted preferred, MS-assisted allowed, MS-based preferred and MS-based allowed.

Apart from RRLP, it should be noted that due to the limited adoption of EOTD, emergency services primarily utilize UTDOA (3GPP-TS-43.059) (Uplink Time Difference Of Arrival) in GERAN networks. Moreover, the actual assistance data requests are not delivered in RRLP, but in the BSS Application Part LCS Extension protocol (3GPP-TS-49.031).

3GPP TS 25.331

Radio Resource Control (RRC) 3GPP TS 25.331 (3GPP-TS-25.331) is the radio resource control protocol for the User Equipment (UE) - UTRAN interface. RRC defines, amongst other items, similar functionality for positioning of an UE in an UTRAN network as RRLP does for positioning of an MS in a GERAN network. It should, however, be noted that whereas RRLP is a standalone positioning protocol with termination points at MS and SMLC, RRC carries in addition to positioning payload also a plethora of other data. Hence, RRC is not only a standalone positioning protocol. RRC is terminated at UE and RNC (Radio Network Controller) of the UTRAN. RRLP and RRC therefore differ in scope and implicated architecture, although both can carry the same type of positioning and location information.

In addition to AGNSS-based positioning, RRC also provides a RAN-based trilateration method called IPDL-OTDOA (Idle Period DownLink - Observed Time Difference Of Arrival). Similarly to EOTD, IPDL-OTDOA requires infrastructure investments and, hence, the deployment has been limited.

3GPP2 C.S0022-A

C.S0022-A (or IS-801-A) (3GPP2-C.S0022-A) defines a position determination protocol for IS-95/IS-2000 and HRPD (High Rate Packet Data) systems and is maintained by 3GPP2. The capabilities of C.S0022-A are similar to its 3GPP counterparts. The support for additional satellite systems will be included in the coming release C.S0022-B.

The IS-95/IS-2000 networks also support TOA-based (Time Of Arrival) positioning method called Advanced Forward Link Trilateration (AFLT). AFLT is based on the time synchronized base stations that allow the network or the terminal to calculate the position estimate of the terminal based on the TOA measurements.

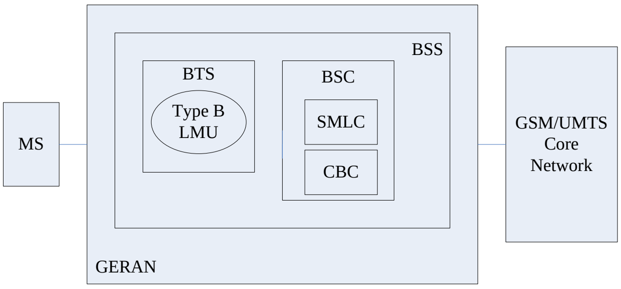


Fig. 1 GERAN LCS architecture.

OMA SUPL 1.0 and 2.0

The previously considered RRLP, RRC and TIA-801 are positioning protocols for Control Plane of the cellular networks - they are an integral part of the cellular network. However, in addition to Control Plane solutions, there are also User Plane solutions, which provide assistance and positioning data over IP-networks.

Examples of User Plane solutions are the SUPL (Secure User Plane Location protocol) Release 1 (OMA-TS-SUPL-1-0, 2007) and (draft) Release 2 (OMA-TS-SUPL-2-0, 2009) standardized in OMA (Open Mobile Alliance). SUPL architecture provides a wide-range of services, such as authentication, security and charging, through other enablers (defined by OMA, 3GPP or other standardization fora) as well as various location services including triggered periodic and area events. Therefore, the OMA LCS architecture with SUPL can be considered to be a complete end-to-end solution as required of OMA enablers.

In positioning technologies OMA SUPL relies on Control Plane protocols, such as RRLP and RRC, which the SUPL encapsulates as sub-protocols (see Fig. 2). Over the recent years the importance of SUPL has increased due to the growth in the LBS business. Increasingly the primary use for the Control Plane methods, such as A-GPS as well as AFLT in the CDMA networks and U-TDOA in GERAN networks, is in emergency services, whereas LBSs are based on User Plane positioning solutions.

Fig. 3 introduces a simplified OMA Location Architecture - for full architecture see (OMA-AD-SUPL-2-0, 2008). The architecture shows the major entities including SUPL Location Platform (SLP), Short Messaging Service Centre (SMSC), WAP PPG (Wireless Application Protocol Push Proxy Gateway) and SET (SUPL-Enabled Terminal), which is the terminal to be positioned. The functional entities of SLP, SUPL Location Centre (SLC) and SUPL Positioning Centre (SPC), handle amongst other items subscription, authentication, security, charging, privacy, positioning and assistance data delivery.

In the SUPL framework the positioning session can either be SET-initiated or Network-Initiated. In the SET-initiated case the SMSC and WAP PPG do not have a role, but the SET directly connects to the SLP (in proxy mode – the behaviour in the non-proxy mode in CDMA networks is somewhat different) and, for example, retrieves the required assistance data from the SLP. In the Network-Initiated case the SET must be notified so that it knows to set up data connection to the SLP. The channels to deliver the notification are, for example, over an SMS (text message) or over WAP. In an exemplary case of the network-initiated session a SUPL Agent external to the SET (an application, for instance) requests SLP to position the SET. Having received the request the SLP sets up a Network-Initiated session with the SET using an SMS and positions the SET.

The advantages of OMA SUPL lie in the possibility to rely on other OMA enablers and also on other standardized architectures including SMS. The Network-Initiated sessions can, for example, be utilized in various services as well as in lawful interception and positioning of emergency calls.

LTE and WiMAX considerations

The emerging RANs, 3GPP E-UTRAN and WiMAX (based on IEEE 802.16), also need positioning solutions. The 3GPP LTE has a work item open for an LTE-native Control Plane solution that will incorporate AGNSS as well as time difference –based methods. WiMAX Forum has agreed to use SUPL as one option for positioning. Moreover, in WiMAX Forum there is also a work item towards a WiMAX-native positioning solution called WLP (Wireless Location Protocol).

It should be noted that the (draft) SUPL Release 2 supports both LTE and WiMAX and, hence, the use of SUPL might be an adequate solution in these networks. The draft release also supports LTE-native positioning protocol, even though it has not been defined yet. However, its inclusion can be justified by respective timelines of future SUPL releases and LTE Release 9.

RRLP	RRC	LTE RRC	TIA-801
ULP			
TLS			
TCP/IP			

Fig. 2 OMA SUPL Release 2 protocol stack.

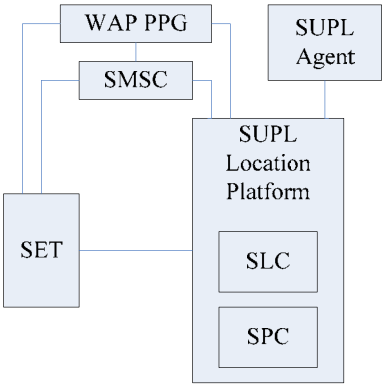


Fig. 3 OMA Location Architecture

3. Shortcomings in the Existing Positioning Protocols

The Control Plane positioning protocols have been developed and evolve based on RANs' needs (mainly emergency call positioning) and capabilities. The positioning protocols defined in 3GPP and 3GPP2 contain RAN-specific items that are not needed outside the scope of the respective RAN. These include, for instance, the native RAN methods including EOTD and AFLT. Hence the use of RAN-specific positioning protocols complicates the User Plane deployments. This is especially true of RRC, which is the protocol for the radio resource control in general.

Moreover, the Control Plane protocols suffer from the limitations of the protocols lower in the hierarchy in the RAN protocol stack and prevent realization of novel features. For instance, in the Control Plane it has been impossible to realize solutions for high-accuracy sub-meter positioning methods, such as Real-Time Kinematic (Leick, 2004) because of bandwidth, architecture and protocol limitations. To be more specific, for instance RRLP is designed for one-time point-to-point assistance and measurement delivery and is, therefore, unsuitable for positioning methods requiring constant stream of reference measurements (Wirola et al., 2007b and 2008b).

Because the Control Plane protocols are also being utilized in the IP-networks via the use of OMA SUPL, the in-built protocol limitations have also been copied to the User Plane solutions. This leads to sub-optimal solution, because in the User Plane the bearer-networks and -protocols would in fact be capable of providing more services and significantly larger bandwidth for positioning purposes.

Considering the current and future needs one of the most serious flaws in the 3GPP-based protocols is the lack of support for the signal-of-opportunity -based, such as WLAN, positioning. The Control Plane protocols do include the support for the RAN-native network measurements, but they lack the capability to transfer measurements made from other RANs or radio networks. For instance, because IEEE networks are out-of-scope of 3GPP, it is highly unlikely that 3GPP would define positioning methods that are based on IEEE technologies including WLAN. The same also applies vice versa.

In the (draft) SUPL Release 2 this deficiency has been overcome to some extent by incorporating items for signal-of-opportunity positioning in the ULP-layer (User plane Location Protocol) of SUPL shown in Fig. 2. These capabilities include the possibility to report radio network measurements from various networks, including GSM, WCDMA, LTE and WLAN in the ULP-layer messages.

The ULP-layer defines messaging, for example, for initiating and terminating the SUPL session as well as capabilities handshakes and service subscriptions. However, because the AGNSS and the other RAN-based positioning methods (EOTD, IPDL-OTDOA) are encapsulated in the sub-protocols to the ULP, the layer-approach typically adopted in the protocol design is dismantled due to the positioning technology additions made to the ULP layer. Therefore, the structure and capabilities of SUPL have suffered significantly from inheriting the limited Control Plane positioning protocols. Hence, also OMA SUPL would benefit from developing a flexible and comprehensive positioning technology protocol solely for the User Plane.

Another challenge in the user plane is the GNSS fine time assistance. The more accurate the time assistance is the more precisely the AGNSS receiver can predict the Doppler and code phase in order to improve sensitivity and, hence, the time to first fix. In the Control Plane the GNSS time assistance is tied to the cellular frame timing. However, as mentioned, this requires the deployment of LMUs in the network. The same capability is also available in the User Plane. However, this approach has a drawback that it makes SUPL inherently operator-tied service, because access to the core network is needed in order to obtain the GNSS-cellular time relations.

In the User Plane alternative solutions to be considered include Network Time Protocol (NTP) and timing services available in the Internet. However, the latency of the IP-network, especially over the air, results in unpredictable errors in the time assistance. Another future option is to obtain the cellular time -tied time assistance from another terminal in the same cell over a peer-to-peer network or via a server caching the cellular timing data collected from terminals. However, in any case these latter methods would not provide a solution for, say, WLAN-only devices. Therefore, time assistance over the IP-network remains a challenge.

Finally, one of the drawbacks in performing positioning in the User Plane using OMA SUPL is that WLAN-only devices cannot utilize all the SUPL services unless the terminal and the WLAN network are I-WLAN -enabled (3GPP Interworking WLAN). For example, authentication in OMA SUPL requires having a SIM card (Subscriber Identification Module) in the terminal and a subscription to the 3GPP network. I-WLAN provides a mechanism to support 3GPP specified mechanisms, including authentication, over the WLAN bearer.

4. Trends in location technologies

Positioning services can be characterized by four attributes: availability, accuracy, integrity and authenticity of the source. Availability refers to the fraction of time, when positioning is possible. For example, GNSS-based positioning has excellent availability in rural outdoor conditions. However, in urban and indoor environment the availability degrades rapidly.

Accuracy, on the other hand, refers to how precise location information a given positioning technology may yield. Typically GNSS is considered an accurate technology, whereas cell-based methods are referred to as inaccurate technologies with a potential position error of several kilometres.

Integrity refers to the reliability of the positioning service. For instance, in GNSS-based positioning integrity may be compromised by a faulty satellite. Because of this satellites send their health (or integrity) data to the user equipments. The integrity information is also provided in the AGNSS assistance.

Finally, authenticity refers to the authenticity of the signal source. Typical examples for signal authentication include the methods to prevent the spoofing of GNSS signals. Spoofing can be understood to mean misguiding of users by means of forged signals (Günter, 2007). While military users have always been concerned with the potential spoofing and jamming of the signals, these aspects are also of growing importance in the civilian sector now that, for instance, location-based security solutions are being introduced. Moreover, in addition to deliberate forging attempts, unintentional interference from in-device or from other devices are potential sources of errors.

While integrity and authenticity are major concerns in the emergency services, they are not currently considered as major drivers in developing positioning technologies for location-based services. This is due to the inherent problems with availability and accuracy in consumer solutions, such as positioning services in mobile terminals. These issues must be solved first. However, as technologies develop in these areas, solutions in the areas of integrity and authenticity will be required as well. For instance, applications requiring or providing location-based charging necessitate integrity and authenticity guarantees. One option to tackle both spoofing and interference is to have at least two independent positioning technologies enabled in the device.

Therefore, the two near-future driving factors in the location technologies are accuracy and availability. Accuracy requirements can be addressed by enabling more advanced GNSS-based positioning methods and AGNSS assistance to the consumers. From technology point-of-view it would be possible to provide the end users with high-accuracy GNSS positioning methods, such as Real-Time Kinematics (RTK) (Wirola et al., 2006) and Precise Point Positioning (PPP) (Leick, 2004). However, these methods both require new protocol messaging as well as new types of assistance data services, such as high-accuracy navigation models as well as regional atmosphere models in the case of PPP. These methods cannot be realized in the Control Plane protocols and extending SUPL to sew up the sub-protocol (RRLP or RRC) shortcomings has been shown unfeasible (Wirola, 2008b).

Although a full-scale multi-frequency RTK may be an overkill for a handset integrated GNSS, it is feasible to realize at least a light-version of RTK using an external GNSS-receiver connected via Bluetooth to the device (Wirola et al. 2006 and 2008a). The increasing availability of satellite systems and civilian signals in consumer-grade GNSS devices will eventually enable the technologies now in professional use also to the wider audience. By the light-RTK the authors refer to abandoning rigorous integrity requirements of professional RTK solutions to some extent and also on being satisfied with a float solution.

Moreover, the availability of GNSS reference networks and, hence, the availability of virtual reference measurement services introduce interesting opportunities for future high-accuracy positioning technologies for consumers. However, in order to realize this potential the standardized positioning solutions must be able to carry appropriate data content, which they are not capable of doing at the moment.

The same also applies to PPP. A rigorous professional-quality PPP may not be feasible for consumer devices, but significant performance improvements can already be achieved by enabling high-accuracy navigation models and, say, regional troposphere and ionosphere models. Again, such a PPP solution might be called light-PPP.

The discussed high-accuracy AGNSS methods, however, have low availability due to the requirement to have good or excellent satellite signal conditions. The availability aspect, on the other hand, can be addressed by the radio network -based methods based on fingerprinting, fingerprint databases and associated positioning technologies.

A fingerprint database is defined as a grid, in which each grid point is associated with a set of measurements from a set of radio networks (Honkavirta, 2008). The measurement types include time delay measurements, time difference measurements between the base stations, channel or signal quality measurements (power histograms, number and spread of RAKE fingers, pulse shapes) and measurements from multiple antennas (diversity receiver). The databases may have wide or even global coverage.

An important aspect in signal-of-opportunity –based positioning is that it must be based on existing infrastructure. Limited areas, such as hospitals, can be populated with special positioning tags or similar, but a global scale positioning solution must take advantage of already existing wide-spread infrastructure. This can be seen as one of the drivers for WLAN-based positioning. The WLAN infrastructure is widely available and various devices are already equipped with a WLAN chip. Hence its utilization in positioning is a natural step. The only remaining aspect is the availability and transfer of WLAN access point maps, which transfer is currently in the scope of no positioning standards.

The IEEE 802.11 has activity towards standardizing an interface that allows the access point to report its position to the terminal or vice versa (IEEE, 2008). However, it takes time to replace the existing WLAN infrastructure with new equipment supporting new standards. And even then, not all the access points may have their coordinates set for further distribution. Hence, the current WLAN-based positioning solutions rely on databases with records of access points versus their coordinates in the simplest form of databases.

As discussed, the fingerprint positioning solutions almost completely lack support in the location standards. Although the (draft) SUPL Release 2 supports reporting GSM, WCDMA, LTE, CDMA, HRPD, UMB (Ultra Mobile Broadband), WLAN and WiMAX network information, SUPL is not designed for the fingerprint collection. Moreover, OMA SUPL is based on an assumption of network-based SET-assisted RAN-based positioning and, hence, it is impossible to transfer a fingerprint database to the terminal for positioning purposes using the current SUPL versions. Again benefits for SUPL can be seen in defining a new positioning technology package for the User Plane LBS needs.

Moreover, the support for sensor-generated measurements and information originating from e.g. accelerometers, magnetometers and barometers is not covered by the current standards. For instance, although heading information is supported in various standards, motion state (walking, running, etc.), which can be extracted from the accelerometer data, is not. Moreover, taking advantage of the full potential of barometers requires availability of either pressure reference data or troposphere models. Sensors are also expected to play a major role in addressing the indoor positioning challenge (Alanen et al., 2005).

Due to the limitations in the currently utilized standardized Control Plane/User plane solutions several User Plane -oriented proprietary systems have been developing in the market. Examples include WLAN-based positioning solutions as well as proprietary GNSS assistance data services. These are differentiators in the market and all the techniques can never be standardized due to the intellectual property right and business secret issues. Although the standards cannot provide unified interface towards these services, the standards could still, however, provide generic containers for proprietary payloads so that both, standardized and proprietary assistance, could be carrier within the same standardized framework. The advantage of such an approach is that each new assistance service would not then have to define a new protocol.

The introduction of proprietary containers would also work to prevent the fragmentation of the positioning protocols. Currently each new RAN is forced to define a new positioning protocol for its native use. In addition, each new assistance service is compelled to define a proprietary protocol for carrying the data. The negative effects of such fragmentation include increased costs due to the need to support multiple protocols.

Yet another driver for the location technologies is the location-awareness and power consumption, which are closely related. Being location-aware requires performing positioning periodically or based on some other criteria such as change of an area. However, such frequent positioning events lead to increased power consumption and also to data costs. Hence, the location technologies being developed generally try to minimize the data connections – an example of such are predicted ephemeris services. Moreover, such power consumption requirements also lead to positioning being performed by the technology that just and just fulfils the required quality-of-service. For example, if only crude position estimate is required, GNSS shall not be used. Instead, the terminal is always aware of its serving cell and, hence, assuming an availability of an appropriate fingerprint database, the terminal can be positioned without significant additional energy consumption.

Also, the applications utilizing location data, such as Nokia Maps, operate on the User Plane and are becoming more interactive. Therefore, it is natural that the development of the location technology protocols is concentrated in User Plane, not in the Control Plane.

In conclusion, the discussion above shows that there is a need for standardization activity in location technology protocols in the User Plane. Authors have been proposing a work item for the LTP (Location Technology Protocol) in the OMA Location working group, but so far such a work item has not been approved.

5. Use Cases

In the current location solutions (for example see Figs. 1 and 3) there is a strict architecture with the location server (for example SMLC in GERAN and SLP in SUPL) providing the terminals (MS or SET, respectively) with assistance and positioning instructions. The network element has been given the control of the positioning session - it is the network element that decides, or recommends, which positioning method shall be used. These both issues must change since they imply heavy architecture and network-controlled positioning session, respectively.

Firstly, the LTP must not limit the roles of different entities – instead, the LTP can find its use between various different types of entities. Any entity (for instance, handset, laptop, server and service/data provider) can work in any role. For example, traditionally there has been a server providing terminals assistance data – however, it is equally feasible for the server to request assistance data from terminals for distribution to other terminals.

Similarly, the LTP messaging must also flow between any types of entities. For instance, in device-to-device relative positioning measurement messages are exchanged between two devices, for example two terminals, not between a terminal and server. This implies that the LTP must not be tied to any specific architecture, because any entity can request and deliver almost any data.

Such a concept is shown in Fig. 4, in which different entities are represented as nodes that are termination points of the LTP. Any node should, in principle, be allowed to work as a data producer (i.e. allowed to publish, for example, the satellite ephemerides the node has received) in the location network. Also, any node should be able to function as a data provider (i.e. work, for instance, as a cache server for assistance data) in the network. Such a scheme opens up a possibility to set up community-based assistance networks.

Note that the complexity with, for instance, security, charging, privacy, setting up the point-to-point or even point-to-multipoint as well as multipoint-to-point connections is hidden in the bearer protocol. Such aspects are not in the scope of the location technology protocol, but are taken care by the bearer protocol encapsulating the LTP. The bearer protocol is indicated in Fig. 4 by the notation B(LTP), which refers to the LTP being encapsulated by a bearer (B) protocol.

The retrieval of assistance data from an external source to the location server for distribution has thus far also been out-of-scope of location technology protocols. However, the data from the Wide-Area Reference Networks (WARN) is essentially similar to the data provided by the location server to the terminals. Therefore, the third use case to consider for the LTP is in this interface. The requirement can again be achieved by considering the WARN feed provider as a node (see Fig. 4) with certain capabilities. Cost savings can be induced by standardizing also the channel between the data provider and the assistance server.

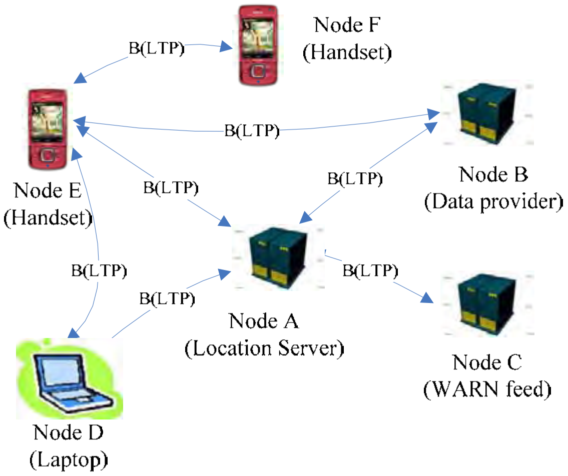


Fig. 4 The LTP is designed to not to limit the flow of information between the nodes in the location network. However, the bearer protocol may limit the actual connections. The notation B(LTP) is introduced to highlight that LTP needs to be encapsulated by a lower level bearer protocol.

The fourth use case to consider is the event-based assistance data. Currently, the protocols are designed so that the terminal either requests assistance data or the server pushes assistance data to the terminal in the beginning of the positioning session. However, there are emerging assistance data types including atmosphere models and long-term GNSS navigation models, which require that a serving node must be capable of pushing updated assistance data to the terminal as the data changes. Also, the nodes must be able to subscribe this data.

Finally, the fifth use case for the data content in the LTP is broadcasting. Majority of, for instance, GNSS assistance data is global by nature. Therefore, server loads and bandwidth requirements can be eased, if GNSS assistance were broadcasted, for example, over OMA BCAST (OMA-TS-BCAST, 2008). Also, some data may be regional. For example, Europe-wide ionosphere maps could be broadcasted using OMA BCAST, because the enabler also provides means to control the distribution geographically.

Also RTK measurements from GNSS reference networks are suitable for broadcasting. In such a case measurements and locations of the reference stations are distributed so that the terminal can process all the measurements (from the network and the terminal) in a single filter. The approach has been shown to produce superior results (Dao, 2005). Another option is to distribute reference measurements and spatial correction terms to the terminals. Finally, also updates to the geographically-segmented fingerprint databases could be delivered over the OMA BCAST channel.

6. Data content requirements

In the Assisted GNSS -side, the LTP must offer the same types of AGNSS assistance as today - for thorough discussion about AGNSS assistance refer to (Syrjärinne et al., 2006). This includes being able to provide data common to all the GNSSs (such as, ionosphere model) and GNSS-specific data (such as navigation models) in a generic format. Also, an important aspect is having a multi-mode navigation model enabling providing GNSSs navigation models also in non-native formats (Wirola, 2007a). The present protocols also support differential GNSS, data bit assistance, earth-orientation parameters and real-time integrity. All these must be supported by any subsequent protocols. In general, the current positioning standards for AGNSS support effectively all the content available in the GNSS broadcasts.

In addition to the broadcast data types, the data must also support reference location and time. Reference location may be given based on radio network –data. The reference time must be defined in such a way that it can be given with respect to any given radio system. Currently the RAN-specific positioning standards support only giving GNSS time with respect to the specific RAN time. However, the frame timings that are typically given in RAN-specific units for reference time purposes can be reduced into common units including SI-units.

Also, the future protocol must have suitable content to support novel high-accuracy GNSS positioning methods. This means having certain measurement types, namely code phases and carrier phases at suitable resolution, in the standard (Wirola et al. 2007b) in order to be able to support RTK. For PPP the new required data content includes high accuracy navigation models, differential code biases and regional atmosphere (ionosphere, troposphere) models. Additionally also, for instance, antenna information may be considered.

Finally, the AGNSS side must also consider the emerging predicted navigation model services and their derivatives. The 3GPP specifications already include one implementation of predicted navigation models that can provide the terminal navigation model data for several days ahead. However, there are also other implementations and also data transfer needs for proprietary services including autonomous predicted ephemeris generation in the terminal.

In the radio network -based positioning the assistance to be carried by the LTP consists of fingerprint database. These items are not consistently included in any location standard. Although the (draft) SUPL Release 2 specification defines measurement parameters for a number of networks, the parameters are not equal between the systems (contents of data elements, resolutions, ranges). The hybrid use of different networks is, therefore, very difficult due to profound differences in the measured parameters and the measurement report contents in the User Plane specifications.

The generic fingerprint to be included in the LTP must therefore equalize the systems by providing, for example, such generic timing (or time difference) and observed signal strength measurement report that it is applicable to all the systems. Only then are the real hybrid methods feasible. This addresses especially the availability challenge. The systems considered may include GSM, WCDMA, WLAN, WiMAX, Near-Field Communications, Bluetooth and DVB-H (Wirola, 2008c).

The positioning using the fingerprint database is based on statistically comparing the measured fingerprint to the database records (Honkavirta, 2008). Another type of data, based on fingerprint database however, suitable for positioning are radiomaps (Wirola, 2009) that contain access point and/or base station coverage area models in terms of shapes defined in 3GPP GAD (3GPP-TS-23.032) including ellipses, ellipsoids or polygons.

Finally, the support for sensor-generated measurements and information originating from e.g. accelerometers, magnetometers and barometers must be covered. For example, supporting barometer fingerprints could allow the server to keep its pressure assistance grid up-to-date for assistance data purposes instead of relying on weather forecasts or similar.

Finally, the possible broadcast of the data elements over the OMA BCAST introduces no additional requirements for data coding. The data content carried within the OMA BCAST can be anything, for instance a file. However, the broadcasting possibility should be borne in mind, when defining the data content so that, for instance, geographical applicability aspects are adequately taken into account.

7. Protocol stack requirements

Fig. 5 shows the schematic protocol stack used with the Location Technology Protocol, which is the highest protocol layer. It handles all the positioning-related messaging and data transfer.

In addition to the LTP, a lower level protocol is required to handle transporting the LTP payload from one node to another simultaneously handling, for example, user authentication, security, privacy and charging issues, if required. This protocol encapsulating the LTP is called the Routing Protocol in Fig. 5. This can be thought to be the bearer protocol indicated in Fig. 5.

The protocol requirements to the LTP itself include that it must be capable of error handling and recovery – a typical situation with AGNSS assistance is that the entity providing assistance data cannot provide all the data the terminal requested. The protocol must therefore not expect to get all the requested data, but be capable of a re-requesting other assistance data, if applicable. For this purpose the termination points must also be able to exchange their capabilities, namely to report what their positioning method and assistance data capabilities are.

Furthermore, the LTP messaging shall be symmetric so that the LTP does not imply that it is always the terminal that requests assistance data – equally well a server may request assistance data from some entity. Symmetric messaging, therefore, enables abandoning the current scheme of strict division between the MS and Location Server.

While version control is a natural requirement of any protocol, the LTP is also envisioned to be stateless in order to maintain the scalability of the infrastructure that includes, for example, multiple servers. However, depending upon the services provided the Routing Protocol may need to have states if the deployment supports, for instance, sessions for continuous periodic exchange of measurements (streaming). This is required, for example, in RTK that requires a possibility to request and deliver a stream of measurements from one node to the other. The streaming is then realized in the Routing level and the exchange of measurements in the LTP layer.

The lack of states also means abandoning the conventional methods such network-based MS-assisted mode, in which the network orders the MS to take measurements and return them to the network for position determination. Giving up such schemes is natural, because the terminal capabilities have increased and terminals are nowadays fully capable of performing, for instance, all the calculations required for position determination. Hence, the role of the network (server) side should be more supporting than imperious.

The Routing Protocol may either be a very simple or arbitrarily complex one depending upon the services it must provide. However, the possibility to have a simple Routing Protocol, which in its simplest form only need to open a data pipe between two nodes, serves research and development work as well as academics. In certain deployments this also yields cost advantage. The realization of the Routing Protocol in each deployment ultimately depends upon the environment.

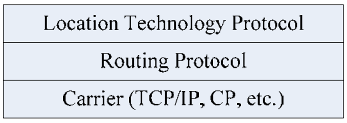


Fig. 5 Schematic protocol stack

Note that the definition of this bearer protocol is tied to the architecture. In an exemplary case in Fig. 5 the Node A may work as a master node (location server) to which all the other nodes register with their capabilities. A node might for instance register with a capability that it can provide the other nodes broadcast A-GLONASS assistance limited to the satellites visible to the node. Therefore, in addition to registration, the bearer protocol must, in this case, provide the means to route, say, assistance data requests originating from one node to another node capable of providing the assistance data.

Now, the architecture defined in this example sets requirements to the bearer protocol, but the underlying LTP is unchanged. The adaptation is, hence, in the Routing level and is transparent to the LTP. This is essential from the protocol transferability point-of-view. It is also the Routing protocol that limits the roles of the entities based authentication, security privacy and charging requirements.

Although the Routing Protocol is out-of-scope of the LTP and this article, it should be recognized that different features provided by the LTP require different levels of service from the Routing Protocol. It is therefore advisable to categorize the LTP features into service packs. The service pack definition consists of the subset of the LTP features and of the requirements their implementation sets for the Routing Protocol.

8. Exemplary implementation

In the 3GPP systems the Routing Protocol already exists to some extent, because for example authentication is inherent to the architecture.

In the IP-networks the Routing Protocol could be the ULP, because it can rely on, for instance, security (based on TLS, Transport Layer Security), authentication (based on 3GPP GBA (3GPP-TS-33.220), Generic Bootstrap Architecture) and charging mechanisms already defined in 3GPP, OMA and other fora. However, it should be noted that the (draft) SUPL Release 2 cannot support the LTP, but the future releases of SUPL could consider the LTP, if standardized, as a location technology enhancement exerting certain requirements on the ULP-layer as well as on the OMA LCS architecture.

Fig. 6 shows the realization of the LTP in the OMA SUPL architecture. The LTP has been introduced alongside the current 3GPP/2 positioning protocols as a sub-protocol to the ULP. Within the LTP there are modules for different positioning technologies including GNSS and Radio Network –based positioning. The LTP itself contains the capabilities handshake and positioning requests for different positioning technologies or their hybrids.

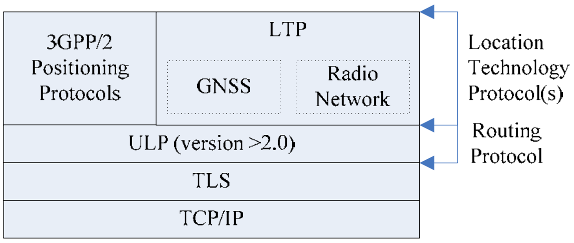


Fig. 6 Exemplary stack implementation

The GNSS module in the LTP is the protocol for Assisted GNSS data. It includes the content, request and delivery mechanisms found in the 3GPP AGNSS specifications. Also included are, for instance, capabilities to request and deliver regional atmosphere models, measurements required for high-accuracy methods and the multitude of non-native navigation models. Moreover, the historical division to MS-based and MS-assisted methods is not required, because similar functions can be realized through simply requesting and providing assistance data and measurements.

The Radio Network –module is, on the other hand, a protocol for transferring generic fingerprints and radiomap data. For example, through this module it is possible to transfer a WLAN access point coverage area map.

The individual modules can be coded in the wanted formats – they can follow the same coding or have different codings. Exemplary codings include XML and ASN.1. XML has the advantage of being flexible and robust as well as easy to debug. The drawbacks include high bandwidth consumption. The situation can be improved by binary XML such as Efficient XML (W3C, 2008). Even though the binary XML typically achieves good compression ratio, ASN.1 with PER (Packet Encoding Rules) encoding is still superior bit-consumption-wise especially, when the unaligned version is used. On the other hand, aligned PER is more efficient to decode, but consumes more bandwidth. However, extending ASN.1 in future releases results in the code being challenging to follow.

The actual choice of the encoding depends on the anticipated environment as well as future needs. For example, it could be argued that the GNSS-based methods and required assistance data elements are well-established and known and, hence, no major future changes are expected to take place. Hence, ASN.1 is the choice for the GNSS package. On the other hand, the Radio Network –package with new and novel fingerprint databases requires flexibility and expandability. Hence, XML might be the choice for that package.

In order to support, for instance, the delivery of basic GNSS assistance data (navigation models etc.) the OMA LCS architecture and the ULP layer need not be modified to a large extent. In principle the only modification required is adding the indication of the support for the LTP to the ULP-layer. The LTP can then be carried in the same container as the 3GPP/2 positioning protocols.

Bigger changes are, however, required, for instance, for streaming of GNSS measurements between two (or more) users. This requires changes to the OMA LCS architecture and additional messaging so that one user can request such a data pipe to be opened, a network-initiated method to request such measurements from the other user as well as an architecture enabling such routing of measurements for a pre-defined time.

9. Conclusions

Several shortcomings in the currently utilized positioning protocols have been identified in the view of the future location and positioning needs. The current approach of each standardization forum working with location technology protocols for their own domain leads to continued fragmentation of location technology standards and to domain-specific implementations. These domain-specific standards differ in scope and capabilities depending upon the bearer network capabilities as well as the development cycles. Therefore, harmonized positioning performance cannot be guaranteed across all the networks and access network handovers. Moreover, due to the long development cycles of standards, various proprietary location technology protocols have been developing in the market leading to further fragmentation.

Instead, the domain-specific items must be addressed in a lower level adaptation protocol, which is transparent to the location technology protocol. For the location-based services it is important that the location experience is independent of the access network. Such a location technology protocol free of domain-specific hooks can address the needs of every domain (IP, RAN) and lead to convergence in location standards by being re-usable in every domain.

The location technology protocol itself must address in their entirety positioning procedures, messages, measurements and assistance for GNSS-, sensor- as well as radio network -based positioning methods. The protocol must also be as flexible and comprehensive as possible so that additions can be made in fast schedule, when needed. Also, placeholders for proprietary extensions reduce the need for the proprietary protocols in the market.

The authors see that there is a market demand for a comprehensive standardized location technology protocol for User Plane needs. In the long term the standardized solutions are the most cost effective approaches and lead to the widest adoption of the technologies.

Acknowledgements

The authors wish to thank Tommi Laine (Nokia Inc.) and Kimmo Alanen (Nokia Inc.) for providing insightful comments.

References

3GPP-TS-23.032 ***Universal Geographical Area Description (GAD)***

3GPP-TS-25.331 ***Radio Resource Control (RRC) protocol specification***

3GPP-TS-33.220 ***Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture***

3GPP-TS-44.031 ***Radio Resource LCS (Location services) Protocol (RRLP)***

3GPP-TS-43.059 ***Functional Stage 2 Description of Location Services (LCS) in GERAN***

3GPP-TS-49.031 ***Location Services (LCS); Base Station System Application Part LCS Extension (BSSAP-LE)***

3GPP2-C.S0022-A ***Position determination service for CDMA2000 Spread Spectrum Systems***

Alanen, K. and Käppi, J. (2005) ***Enhanced assisted barometric altimeter AGPS hybrid using the Internet***. In Proceedings of ION GNSS 2005, 13th-16th September, Long Beach, USA.

Canalys (2008). ***Mobile Navigation Analysis***. Issue 2008.04 / 9th December 2008

Dao, T.H.D (2005) ***Performance evaluation of Multiple Reference Station GPS RTK for Medium Scale Network***. Master of Science thesis, University of Calgary.

Günter, H. and Kneissl, F. and Ávila-Rodríguez, J. and Wallner, S. (2007) ***Authenticating GNSS, Proofs against Spoofs.*** InsideGNSS, July-August 2007.

Honkavirta, V. (2008) ***Location fingerprinting methods in wireless local area networks***. Master of Science thesis, Tampere University of Technology.

IEEE (2008) ***802.11k-2008***

Leick, A. (2004) ***GPS Satellite Surveying, 3rd ed.*** John Wiley & Sons.

OMA-TS-BCAST (2008) ***OMA-TS-BCAST_Distribution-V1_0-20081209-C File and Stream Distribution for Mobile Broadcast Services***

OMA-TS-SUPL-1-0 (2007) **OMA*-TS-ULP-V1-0-20070615-A*, *User Plane Location Protocol Release 1.0***

OMA-TS-SUPL-2-0 (2009) ***OMA-TS-ULP-V2-0-20090226-D, User Plane Location Protocol Draft Release 2.0***

OMA-AD-SUPL-2-0 (2008) ***OMA-AD-SUPL-V2-0-20082706-C, User Plane Location Protocol Release 2.0 Architecture Document Candidate***

Syrjärinne J. and Wirola L. (2006) ***Setting a New Standard - Assisting GNSS Receivers That Use Wireless Networks***. InsideGNSS, pages 26–31.

W3C (2008) ***Efficient XML Interchange (EXI) Format 1.0***. W3C Working Draft 19th September.

Wirola, L. and Alanen, K. and Käppi, J. and Syrjärinne, J. (2006) ***Bringing RTK to Cellular Terminals Using a Low-Cost Single-Frequency AGPS Receiver and Inertial Sensors***. In Proceedings of IEEE/ION PLANS 2006, 25th-27th April, San Diego, CA, USA, pages 645–652.

Wirola, L. and Syrjärinne, J. (2007a) ***Bringing the GNSSs on the Same Line in the GNSS Assistance Standards***. In Proceedings of the 63rd ION Annual Meeting, 23rd-25th April, Boston, MA, USA, pages 242–252.

Wirola, L., Verhagen, S., Halivaara, I. and Tiberius, C. (2007b) ***On the feasibility of adding carrier phase –assistance to the cellular GNSS assistance standards.*** Journal of Global Positioning Systems, vol. 6, no. 1, pages 1-12.

Wirola, L. and Kontola, I. and Syrjärinne, J. (2008a) ***The effect of the antenna phase response on the ambiguity resolution***. In proceeding of IEEE ION PLANS 2008, 6th-8th May, Monterey, CA, USA.

Wirola, L. (2008b) ***RRLP Shortcomings.*** Technical report LOC-2008-0385, presented in OMA LOC WG meeting, Chicago, USA, 18th-22nd August.

Wirola, L. (2008c) ***OMA-LOC-2008-0303: Generic Fingerprinting in SUPL2.1***, presented in OMA LOC WG interim meeting in Wollongong, Australia, 19th-21st May.

Wirola, L. (2009) ***OMA-TP-2008-0470R02 Socialization of WID 0181 – Generic Location Protocol 1.0***, provided to OMA Technical Plenary 8th January.

User Level Integrity Monitoring and Quality Control for High Accuracy Positioning using GPS/INS Measurements

Washington Y. Ochieng and Shaojun Feng
Centre for Transport Studies, Department of Civil and Environmental Engineering
Imperial College London, London SW7 2AZ, United Kingdom

Terry Moore, Chris Hill
IESSG, University of Nottingham, Nottingham NG7 2RD, United Kingdom

Chris Hide
Geospatial Research Centre, Private Bag 4800, Christchurch 8140, New Zealand

Abstract

This paper presents research undertaken to develop sensor level autonomous integrity monitoring and quality control to support centimetre level positioning in all conditions and environments as conceived under the SPACE (Seamless Positioning in All Conditions and Environments) project. The basic philosophy for integrity monitoring and quality control is early detection of anomalies which requires monitoring of the entire processing chain.

A number of novel concepts and algorithms are developed including algorithms to deal with special issues associated with carrier phase based integrity monitoring (including integration with INS), a new "difference test" integrity monitoring algorithm for detection of slowly growing errors, and a new group separation concept for simultaneous multiple failure exclusion.

Both real and simulated data are used to test the new algorithms. The results show that the new algorithms, when used together with selected existing ones, provide effective integrity monitoring and quality control for centimetre level seamless positioning in all conditions and environments.

Keywords: RAIM, Difference Test, Group Separation

1. Introduction

Integrity is a measure of the level of confidence in the accuracy of the positioning information supplied by a navigation system. It is vital for liability and safety critical applications such as air navigation and some Location Based Services (LBS). GNSS integrity can be monitored at system level, user sensor level or both. User sensor level integrity monitoring falls under the category of Autonomous Integrity Monitoring (AIM) with a pure stand-alone approach referred to as the Receiver Autonomous Integrity Monitoring (RAIM). When a stand-alone GNSS system is aided by another sensor on the user platform, the monitoring process is commonly referred to as User Autonomous Integrity Monitoring (UAIM).

To date significant research effort has been directed at the development of algorithms and techniques for RAIM based on code phase (pseudorange) measurements. There has also been research to develop variations to RAIM particularly in the cases of sensor integration (Lee et al., 1999). The various RAIM algorithms in the literature are largely the same in principle with the differences mainly being in the selection of test statistics and thresholds (Brown, 1992, 1998).

Although positioning data from pseudorange measurements are suitable for many applications, carrier phase measurements are required for applications that require higher accuracy (at the decimetre level or better). Hence, an equivalent to Pseudorange RAIM (PRAIM) is required for carrier phase based positioning. This is referred to in this paper as Carrier RAIM (CRAIM). Pervan (1998) extended the PRAIM concept to CRAIM by assuming knowledge of integer ambiguities. However, this assumption is not always practical. Other approaches, e.g. by Michalson (1995), Pervan (1996, 2003) and Chang (2001) consider ambiguities as unknown variants in the positioning equations. These methods generate float ambiguities with no attempt to fixing the integer values. Therefore, these methods could be unreliable due to the uncertainty in the ambiguity values. Chang (2001) developed a single difference based CRAIM approach which avoids strong correlation issues of the double differenced observable. Due to the same problem of float ambiguities above and the vulnerability of the single difference to receiver clock drift, this approach could also be unreliable.

Beyond the use the exclusive of the traditional measurements from GNSS (e.g. GPS), further complications arise when GNSS data is combined with data from other sensors (e.g. the INS). For example, most of the existing algorithms for monitoring the integrity of integrated GPS and INS sensors are based on the assumption that the INS provides valid (fault-free) data over short periods (Brown, 2006; Gold, 2004; Lee, 1999; Diesel, 1995). Therefore, the purpose of the GPS/INS integrity algorithm design is to detect and exclude any out of tolerance GNSS faults before correcting the INS. This is to prevent corrupted GNSS data from propagating back into GNSS/INS solution (Gold, 2004). The approach provides the integrity monitoring only for GNSS to guarantee the quality of updates provided to the integrated navigation Kalman filter. However, in reality the INS fails and thus to monitor the integrity of integrated GNSS/INS systems, failures in both GNSS and INS should be detected and excluded. Offer (2006) proposed a qualitative detection method which deals only with outliers, for example, in terms of repeated measurements being outside the sensor's dynamic range, and inertial sensor biases being outside the expected magnitudes.

In addition to the algorithmic weaknesses identified above, there are a number of challenges in the development of user requirements, Failure Modes and Effects Analysis (FMEA), conceptual and theoretical issues, and development of new algorithms. The challenges in development of user requirements include consolidation of services and potential environments, quantification of requirements in terms of performance parameters, and flexibility to carry out sensitivity analysis for detailed performance characterisation. The challenges in FMEA process include the analysis of characteristics, classification and modelling of potential failure modes, identification of common mode failures and 'difficult' failure modes, and FMEA for different system architectures and different data types. The challenges in theoretic issues include justification of assumptions such as error sources being independent and normally distributed, appropriate processing of data correlation and systems coupling. Once potential failure models are specified, the next step is to assess the capability of existing algorithms against the failures, with the aim of identifying any 'difficult' failures. New algorithms are the developed to deal with any 'difficult' failures.

For the development of CRAIM, one of the difficulties is the common processing approach that uses the double differenced observable to eliminate the influence of common errors and mitigate the effect of those that exhibit a degree of spatial correlation. Related issues also include ambiguity resolution and validation; cycle slip detection and repair; potential failures associated with differencing (e.g. problems with the reference satellite used for differencing); potential simultaneous multiple failures (e.g. due to multipath and incorrect ambiguity resolution) and correlation of errors.

The results of the FMEA process above identified errors that grow slowly over time such as clock drift (referred to in this paper as Slowly Growing Errors or SGEs) and simultaneous multiple failures as being the most difficult to detect and exclude. This paper proposes a new algorithm based on the "difference test" concept for the former and a new approach based on group separation for the latter. Furthermore, CRAIM algorithms for both stand alone GPS and integrated GPS/INS systems are developed in the paper. These are presented in subsequent sections below.

2. Methodology

2.1 Difference Test

SGEs are of particular concern in filtering based UAIM because the Kalman filter tends to adapt to them. This results in the positioning solution being contaminated with the consequence of misleading information both in terms of accuracy and integrity. For this reason, early detection of this type of failure is vital. Failure detection in RAIM is based on statistical consistency checks using redundant measurements. There are two different RAIM schemes for use with measurements; snapshot and filtering (Brown, 1996). In the snapshot scheme only the current redundant measurements are used to check measurement consistency. However, in the filtering scheme current and previous measurements are used. In either case, the failure detection algorithms are based on a number of assumptions, the most important of which is that residual errors in the measurements are normally distributed. Failure detection consists of three main steps: the construction of a test statistic; the characterisation of the test statistic and the determination of a threshold to reflect the user requirement (e.g. probability of false alert); and decision making. One of the key features in the design of any integrity algorithm is its sensitivity to various types of failure modes. Unfortunately, neither the snapshot nor filtering methods are designed for detecting SGEs. Current approaches for the detection of SGEs have been shown to have significant weaknesses (Feng and Ochieng, 2007a). Therefore, a new algorithm is needed to deal with SGEs.

Normally GPS exhibits long term stability with normal residual measurement noise. However, in the presence of SGEs, the GPS residuals exhibit a rate of growth. Therefore, a test statistic constructed to reflect whether there is a significant difference between the current residuals and residuals a certain time period ago has the potential to enable the detection of SGEs. The ratio test which is based on the F distribution is normally used to test whether there is a significant difference between two independent variants. Hence, the ratio test may be applicable in comparing a faulty variant with fault-free variant. However, if a ramp error occurs before the test, the ratio test cannot give the correct decision. For a ramp error, the ratio over a fixed time interval converges to one with the increase in time, which gives misleading information. Therefore, a new test based on the difference between the conventional test statistics at different epochs is referred to in this paper as the "difference test".

The new test statistic is expressed as:

$$T_{\Delta t} = \sqrt{SSE_t} - \sqrt{SSE_{t-\Delta t}} \qquad (1)$$

Where, t is current time and Δt is the time interval selected. The corresponding degrees of freedom are denoted as dof_2 and dof_1 respectively.

The test statistic is actually based on a moving window and thus always captures the difference between the two edges of the window.

Clearly, it is imperative to characterise and describe the distribution of the new test statistic. This relatively complex task must be undertaken, for example, to enable the computation of the threshold. The norm of the conventional residual follows a Chi-distribution. Therefore, the test statistic ($T_{\Delta t}$) is in fact the difference of two Chi-distributed variants. Based on the analysis of the difference between mean (μ), standard deviation (σ), skewness (γ_1), kurtosis (γ_2) of two Chi-distributed variants, and simulation, it can be shown that a normal distribution $N(\mu_D,1)$ over-bounds the test statistic ($T_{\Delta t}$). The mean of the normal distribution is expressed as:

$$\mu_D = \mu_d - \gamma_{1d} \qquad (2)$$

Where μ_d is the theoretical mean of the difference of two Chi-distributed variants, $\gamma_{1d} = ((\gamma_{12})^{\frac{1}{3}} - (\gamma_{11})^{\frac{1}{3}})\sigma_d / 2$ is the conservative offset factor with γ_{12} and γ_{11} being the skewness corresponding to dof_2 and dof_1 respectively, and σ_d is the theoretical standard deviation of the difference of two Chi-distributed variants (Feng and Ochieng, 2007a).

Based on the above characterisation of the test statistic ($T_{\Delta t}$) in the "difference test", A decision threshold can then be determined form the normal distribution $N(\mu_D,1)$ by taking account of the required navigation performance (RNP), specifically, the integrity and continuity risk from which the probability of missed detection (P_{MD}) and the probability of false alert (P_{FA}) can be derived.

One important factor in the test statistic constructed in expression (1) is the time interval Δt. The choice of time interval depends on the rate of error growth and the length of the data buffer designed. The slower the error growth rate, the longer the time interval required to detect the error. To detect and identify the rate of SGEs, a multiple window scheme can be used at different time intervals. A comparison of performances of the *difference* and *conventional* test algorithms is undertaken and the results presented in the section on 'field trials and results'.

2.2 CRAIM for GPS

Integrity monitoring algorithms are always coupled with positioning algorithms. The resolution of integer ambiguities is a prerequisite to the achievement of centimetre level positioning using carrier phase measurements from GNSS. If integer ambiguities are resolved correctly, then the CRAIM algorithms are a direct extension of PRAIM. Therefore, the resolution of integer ambiguities is a major issue in CRAIM.

In Kalman filtering based positioning, although not specifically a Kalman filtering task, ambiguity resolution is included within the filter used in this paper, rather than considering it as separate task. This is because tasks such as rearranging the double differenced ambiguity states when the reference satellite changes, removing ambiguity states when they are fixed and updating states with the fixed values are all necessary, and would require additional interfaces to the filter. The LAMBDA algorithm (De Jonge et al, 1996) is used to decorrelate the ambiguities to make the integer ambiguity search more efficient. Ambiguity validation is currently achieved using the ratio test (Teunissen, 2005). Once the ambiguities have been resolved, it is necessary to remove the ambiguity states from the state vector of the Kalman filter, as well as to update the remaining states with the resolved integer values. If dual or triple frequencies are used, a cascaded approach is adopted for the resolution of ambiguities, starting with the long wavelength (e.g. wide-lane) measurements which have ambiguities that are readily resolved, and then moving to shorter wavelengths.

A combination of pseudorange, wide-lane and L1 carrier phase observables is used for the positioning. The pseudorange is the most robust but noisy, and can constrain the position solution to a certain level of accuracy. The wide-lane has a much longer wavelength (e.g. about 86cm for L1-L2) than that of any single frequency carrier. Based on the constraint of the pseudorange solution, it is not difficult to meet the error budget for the wide-lane ambiguity resolution. A more accurate solution can then be determined based on the wide-lane. The better positioning solution enables the L1 ambiguity to be resolved more reliably. Hence, the accuracy of the final positioning solution is effectively determined by the L1 carrier phase measurements.

Although the innovation sequence in a Kalman filter contains information obtained from the previous states, it provides the most relevant source of information for integrity monitoring. It is similar to the residual in the snapshot RAIM method (Lee et al., 1999). However, when the double differenced measurements are used, the property of independence in the measurement noise properties is lost. However, this dependence is a mathematical correlation rather than a physical correlation. The measurement noise and covariance matrices in the Kalman filter are used to account for the correlation.

The proposed CRAIM algorithm employs four tests statistics, a full set and three subsets (formed based on the type of measurements i.e. pseudorange subset, wide-lane subset and L1 subset). These subset test statistics help to identify anomalies either in the L1, or L2 data.

The protection level can be determined using relevant information from the Kalman filter. One way is to use the position estimate uncertainty; the other way is to project the test statistic to the position error.

The elements of the covariance matrix (P) indicate the uncertainty of the state. The first three elements in the state are the position in North-East-Down (NED) coordinates. Therefore, $\sigma_H = \sqrt{P_{11} + P_{22}}$ indicates the horizontal position uncertainty. The position estimate uncertainty based horizontal protection level can be expressed as:

$$HPL_1 = k_H \sigma_H \tag{3}$$

where k_H is the factor that reflect the probability of missed detection which is derived from the integrity risk.

Another way to determine the protection level is to project the test statistic to the position error. A ratio of the position error to the test statistic, referred to as *SLOPE* can be calculated based on the observation matrix of Kalman filter. The method is similar to the conventional RAIM. The projection based horizontal protection level is denoted as HPL_2

To be conservative, the protection levels are determined by

$$HPL = \max(HPL_1, HPL_2) \tag{4}$$

Real data contaminated by simulated cycle slips are used to verify the performance of the algorithm. The details are given in the results section.

2.3 CRAIM for GPS/INS Integration

The GPS/INS integration is based on the Kalman filter as well. However, in addition to adding the inertial sensor error models in the state of the Kalman filter, the input observations to the Kalman filter are the differences between GPS based measurements and INS based predicted measurements. The carrier phase based RAIM for the integrated system is referred to in this paper as CUAIM (Carrier User Autonomous Integrity Monitoring). The construction of the test statistic, the calculation of corresponding threshold, and the determination of protection levels are the same as for the CRAIM algorithm for stand alone GPS.

Figure 1 shows the quality control process for a tightly coupled integrated GPS/INS architecture (Feng *et al.*, 2007b). Both GPS and IMU data are sent to the pre-processing module for data screening. This is followed by a pseudorange based GPS RAIM to detect and exclude potential failures. The reason PRAIM is used separately is because the pseudorange measurement is robust. Furthermore, undetected pseudorange anomalies have a negative impact on carrier phase based positioning. In the PRAIM, the carrier phase measurement is excluded if the pseudorange measurement from the same satellites is excluded. Therefore, only the carrier phase measurements with corresponding acceptable pseudorange measurements are passed on to the CRAIM module. The CUAIM uses the measurements that have passed several tests. This is quality control process facilitates failure detection in either GPS, INS or both.

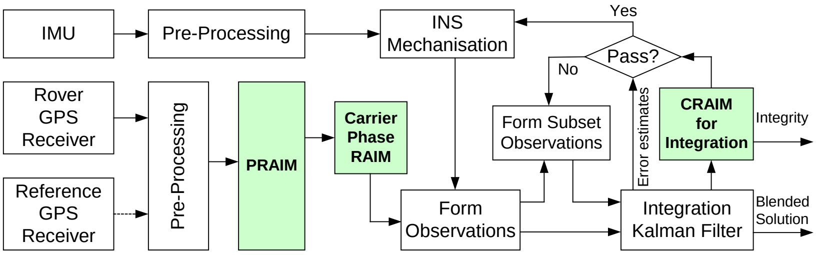


Fig. 1 The integrity monitoring for GPS/INS integration

2.4 Group Separation

Existing RAIM algorithms assume that only one satellite can have a significant error at a time. This assumption is reasonable for some applications such as en-route, terminal and Non-Precision Approach (NPA) phases of flight because the probability of a fault causing a ranging error large enough to cause the position error to exceed the alert limit (e.g. 556m for NPA (or even larger for en route and terminal navigation) is very low (Lee, 2004).

However, at the user level the satellite navigation system is not the only potential source of failure. Failures induced by the operational environment which have the potential to affect several measurements simultaneously have a higher probability of occurrence than satellite (system) related failures. In any case, the one failure assumption of current algorithms does not hold for applications with very stringent requirements such as aircraft landing. For such applications, it is crucial that simultaneous multiple failures are taken in to account. Current methods that extend single failure detection and exclusion to cope with the multiple failures still rely on the assumption of one failure at a time even for a subset. In this case, the failure identification scheme removes one satellite at a time from the full set (all *n* measurements are used) and forms *n* first level subsets each consisting of *(n-1)* measurements. If the failure has not been identified from the first level subset, one satellite will be removed from each first level subset each time to form a bank of second level subsets. Each second level subset consists of *(n-2)* measurements. This process continues until multiple failures are detected and excluded or fails due to either a violation of the minimum required number of measurements, weak geometry or both.

Simultaneous multiple failures are generally of two types. In the first type, independent failures occur at the same time, each causing its corresponding ranging error to become unusually large. In the second type, multiple failures are affected by a common fault (correlated failures) that results in their respective ranging errors becoming unusually large. For the first type of multiple failures, the probability of occurrence is relatively low. Data snooping is probably the only effective way of dealing with these failures. Simultaneous multiple failures are more likely to be of the second type. Therefore, potential common failure modes identified using prior-knowledge of GNSS and the user receiver measurements can be used to determine potential failure 'groups'. The group most likely to fail has the highest priority for separation (exclusion) (Feng and Ochieng, 2006). This approach is referred to in this paper as the group separation method. For example, measurements may be grouped according to 1) navigation system; 2) satellites tracked by the same monitoring station; 3)the age of satellite; 4) satellite clock type; 5) the age of satellite clock; 6) satellite fault/event history; 7) elevation angle; 8) azimuth angle; 9) signal frequency; 10) signal to noise ratio.

3. Field trial and Results

Field trial data combined with simulated inertial sensor anomalies were used to demonstrate the performance of some of the algorithms proposed in this paper: the difference test method, carrier phase based RAIM for stand alone GPS, and carrier phase based RAIM for integrated GPS/INS systems.

The reference GPS receiver used is the Lecia SR530 geodetic RTK receiver which was set to output the pseudoranges (C1, P2) and carrier phase measurements (L1, L2) in the RINEX format. The rover GPS receiver used was the NovAtel OEM4 which was set to output the pseudoranges (C1, P2) and carrier phase measurements (L1, L2) in the RINEX format as well. The Inertial Measurement Unit (IMU) used was the Honeywell's CIMU (Hide *et al*, 2006). The Rover receiver and IMU were mounted on the top of a car.

The trial was carried out on the runway of the Aberporth airport in the UK. The trial route and the position of the reference station are shown in Figure 2. The various equipments on the car were operated in static mode at the start for about 25 minutes. The car was then driven at different speeds along the runway.

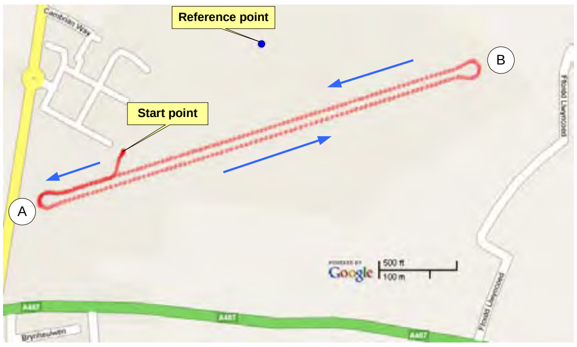


Fig. 2 Trial route (displayed using Google map API licensed for non-commercial use)

3.1 Results of difference test algorithm

Trial data (pseudorange measurements) contaminated by a simulated slowly growing error are used to verify the proposed methods below.

To demonstrate the performance of the algorithms, different time intervals for the "difference test" and different ramp errors start to apply on PRN 21 at the 1500th second from epoch 0. The required probability of false alert is $3.33\times10^{-7}/sample$ (RTCA/DO-229C, 2001). Figure 3 shows the test statistic (difference test) at a time interval of 360 seconds for a ramp error of 0.2m/s. The conventional test statistic is also shown in the figure. The thresholds for the "difference test" are the same if the differences in the number of visible satellites are the same, and vice versa. The comparison of the conventional and the difference test methods shows that the latter is able to detect the failure significantly earlier (by about 20 seconds) than the former.

Figure 4 shows the test statistic (difference test) at time intervals of 120, 240, and 360 seconds for a ramp error of 0.2m/s. It compares the conventional method with the difference test implementing a new early detection

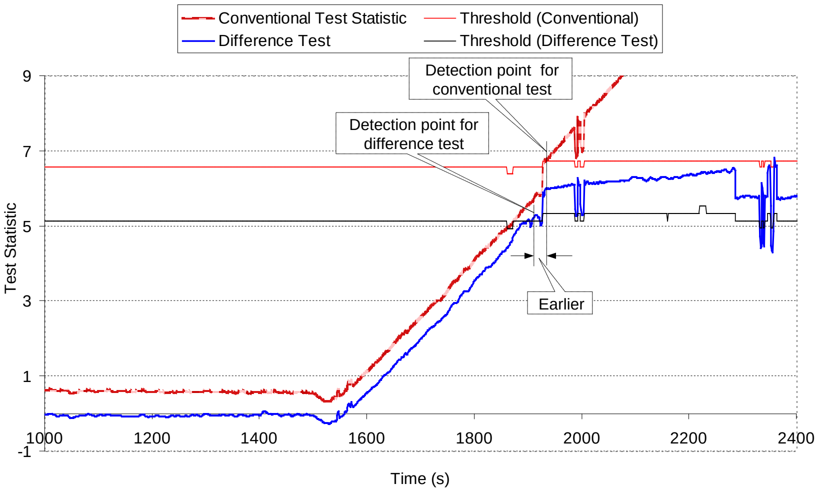


Fig. 3 Comparison of the conventional method and the difference test

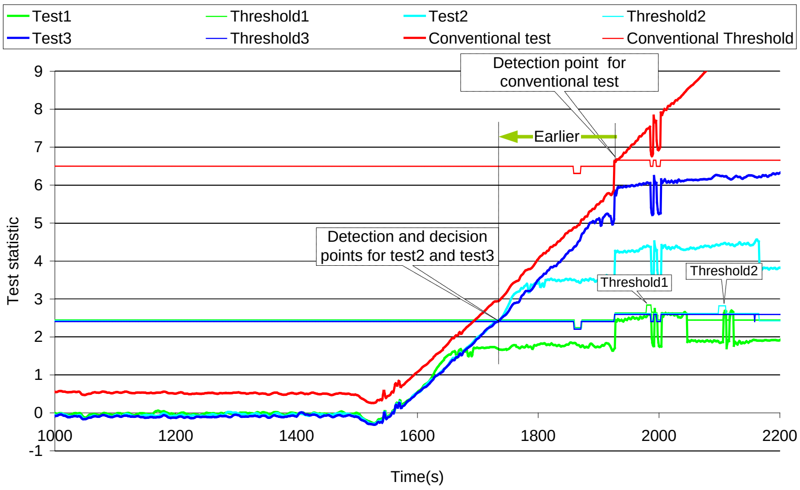


Fig. 4 Comparison of the conventional method and the difference test with the early detection scheme

scheme involving applying the difference test to a number of sequential epochs (Feng and Ochieng, 2007a). The algorithm with the shortest time interval (120s) is not sensitive to this error (test1 in the figure). While the algorithm with the longer time intervals (i.e. 240s and 360s) is sensitive to this error (test2 and test3 in the figure), and can detect the failure much earlier (about 180s) than the conventional method.

This difference test can be used to detect a failure significantly earlier compared to the conventional methods. Results demonstrate that it is able to detect a SGE about 20 seconds earlier than conventional method as shown in Figure 3. This is crucial for safety critical applications such as aviation where the time-to-alert ranges from 5 minutes for the En-route phase of flight to 6 seconds for Category I precision approach with even more stringent requirements expected for Category II and III precision approach (ICAO, 2006). The test statistics based on various time intervals can also be used to identify the rate of growth of error to support FMEA.

3.2 Results of GPS CRAIM Algorithm

Real data combined with simulated cycle slips are used to verify the performance of the CRAIM algorithm proposed in this paper. Based on the assumption that the cycle slips evade the detection in the pre-processing algorithms, the scenario that cycle slip events occur on two L2 carriers at the 200th second is demonstrated. Note that each event involves one cycle slip. The case where the cycle slip occurs on the reference satellite is equivalent to the use of a wrong set of ambiguities. It causes a jump in the test statistics and can be easily detected.

The reference satellite used in the computation of the differences changes at the 785th second in the positioning process. Under each scenario, the test statistic and the corresponding threshold of the full set and subsets are investigated. Figures 5 and 6 show the results of this scenario.

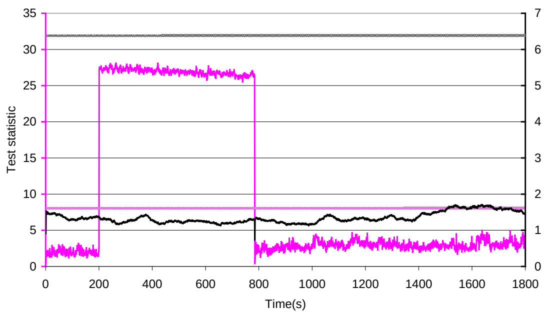


Fig. 5 Test statistics and thresholds for the of full set and the pseudorange subset

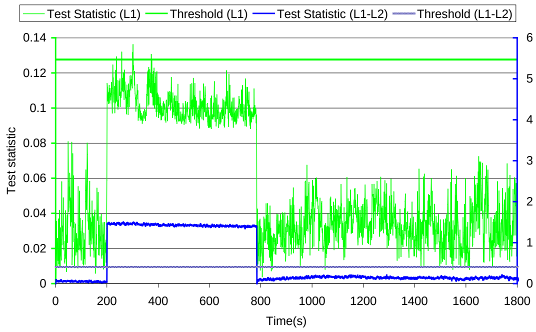


Fig. 6 Test statistics and thresholds for the wide-lane and L1 carrier subsets

The results show that the CRAIM algorithm proposed is able to detect the existence of the effect of cycle slips in L2 measurements in the cases of the full set and the wide-lane subset. However, a few detections occur in the L1 carrier and no detection occurs on the pseudorange subsets as they are not sensitive to the cycle slip(s) on L2. The cycle slip(s) disappears when the reference satellite changes since a new set of ambiguities are resolved at this time.

In the case of the protection level, the dominating factors are geometry and the covariance matrix. There is no significant difference in the protection level for the four scenarios. Figure 7 shows one example result of the protection level. The protection level is relatively stable except at the time (785th second) when the reference satellite changes followed by new ambiguities being resolved in a relative short time (within 2 seconds).This is where the peaks occur in the figure.

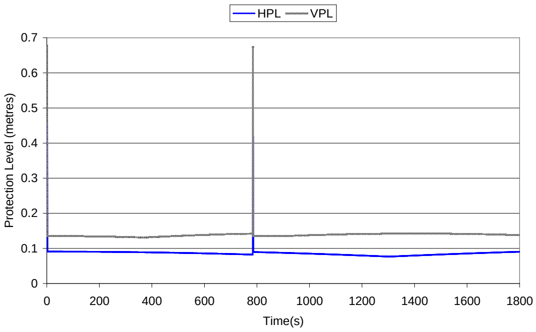


Fig. 7 Example result of protection level

3.3 Results of CRAIM for Integrated GPS/INS

The trial data injected with inertial sensor errors by simulating and adding step errors to the gyroscope and accelerometer data are used to demonstrate two scenarios below:

- Fault-free inertial sensor data
- A step error (1.0m/s^2) applied to the *X* accelerometer at the 45th second from the start of the trial

The position solutions using GPS carrier phase measurements together with the CRAIM algorithm are taken to represent the "true" (reference) trajectory for the analysis of the impact of inertial sensor failure on integrity monitoring. In order to show the results in horizontal and vertical components, the difference between solutions of different configurations is transformed/projected to an East-North-Up (ENU) coordinate representation. Figures 8 to 9 show the differences in positioning results determined from GPS only and integrated GPS/INS without INS failure. The differences in East and North are less than 1 cm in the static mode and less than 5 cm in the dynamic mode. The differences in Vertical (Up) are less than 2 cm. The relatively small differences justify the reasonability of the "truth" assumed above.

Figure 10 shows the test statistic plotted against the threshold. The test statistic is larger than the threshold in a few cases especially around the turning points A and B as shown in Figure 2. This shows the impact of dynamics on the integrity monitoring when inertial sensors are used. The inconsistency occurs not due to failure but due to the varying levels of sensitivity of the GPS receiver and inertial sensor to the dynamics of turning.

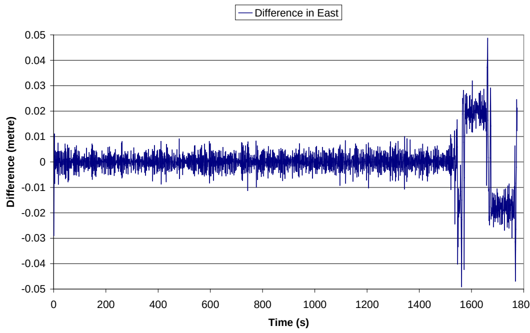


Fig. 8 Difference in East (no failure)

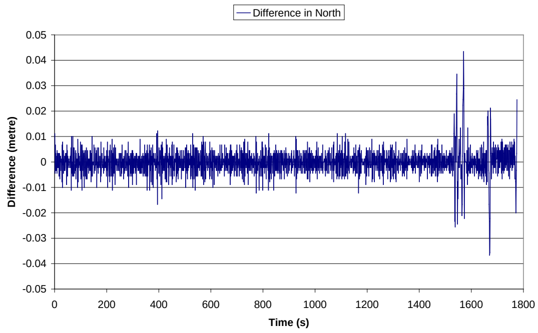


Fig. 9 Difference in North (no failure)

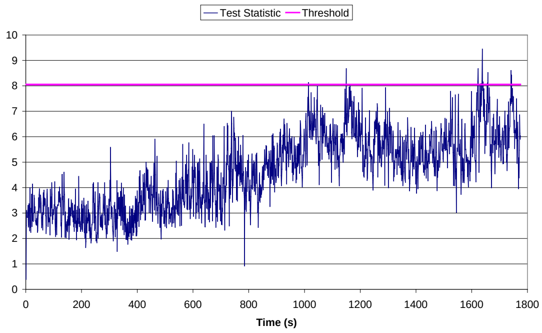


Fig. 10 Difference in Vertical (no failure)

Figures 11 to 12 show the differences in positioning results determined from GPS only and integrated GPS/INS with the X accelerometer being injected with a step error of 1m/s. The differences in East and North at the start of the step are 0.25m and 1.1m respectively. The step error applied to the X accelerometer has a small impact on the difference in the vertical component at the point of introduction of the error. The differences converge to zero over time. This indicates that the Kalman filter is estimating and correcting the X accelerometer error. In the case of the dynamic mode, significant differences occur in all directions especially during changes in the azimuth and pitch. The change in pitch is due to the runway not being horizontal. Figure 13 shows the test statistic plotted against the threshold. The algorithm detects the step error immediately followed by a period of convergence after which the test statistic is less than threshold. However, when the car starts to manoeuvre, the test statistic exceeds the threshold, triggering failure detection.

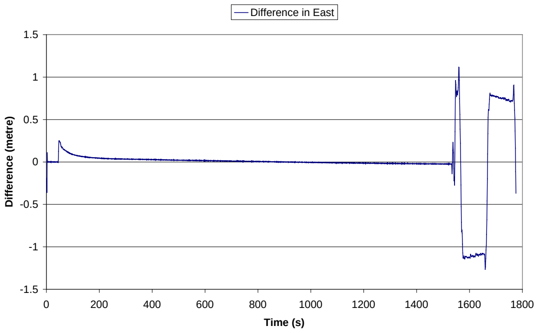


Fig. 11 Difference in East (accelerometer step error)

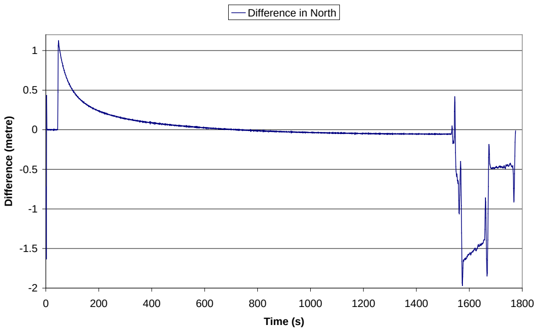


Fig. 12 Difference in North (accelerometer step error)

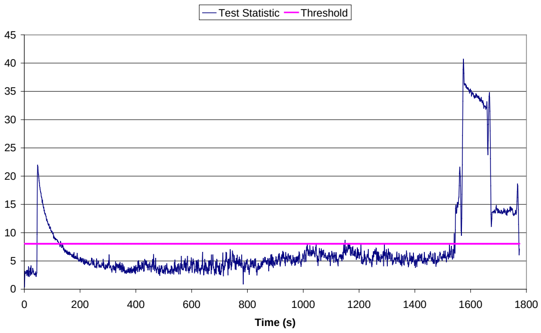


Fig. 13 Test statistic versus threshold (accelerometer step error)

3.4 Results of Group Separation

Simulations results are used below to demonstrate the method based on the assumption of a GPS satellite clock error modeling fault at the master control station which results in all satellites with a cesium clock having a ramp error of 0.05m/s. It should be noted that this example is used for illustration only. Future research will explore more realistic scenarios. The optimised constellation of 24 GPS satellites (RTCA/Do-229C, 2001) and the constellation of 27+3 Galileo satellites are used.

The ramp error is introduced to Cesium clocks starting at 3000 seconds of the week and ending at 6000 seconds of the week. Three satellites in view above a mask angle of 5 degrees are assigned this failure. The snapshot positioning algorithm is used ((Brown, 1996). Both the navigation system grouping and clock type grouping are used in the demonstration. Preliminary results using only pseudorange measurements are shown in Figures 14-16.

The test statistic versus threshold of the combined GPS and Galileo positioning is shown in Figure 14. The failure is detected at around 4300 seconds where the test statistic is larger than the threshold. Figures 3a and 3b show the test statistic versus threshold of the positioning results using the Galileo and GPS group separations respectively. In the GPS solution (Figure 15, where all Galileo measurements were excluded), the failure is detected almost at the same time (around 4300 second) as in the combined solution. While in the Galileo solution (Figure 16), no failure is detected. Combining the results shown in Figures 14 and 16, the conclusion reached is that there is at least one failure in the GPS measurements. Hence, the Galileo measurements can be trusted for positioning. The results show that the group separation method has a reasonable level of efficiency.

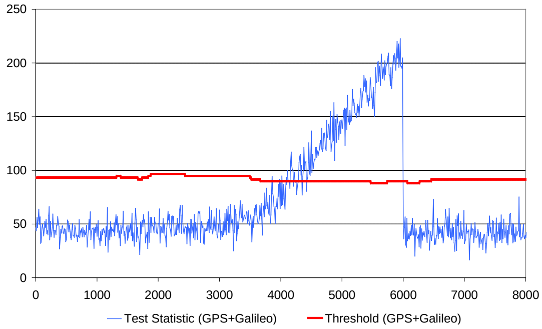


Fig. 14 Test statistic versus threshold for GPS+Galileo with 3 failures

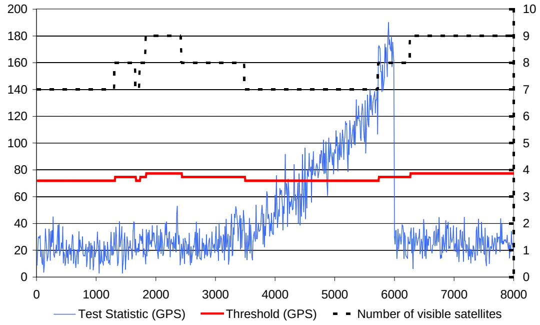


Fig. 15 Test statistic versus threshold for Galileo group separation (i.e. GPS only)

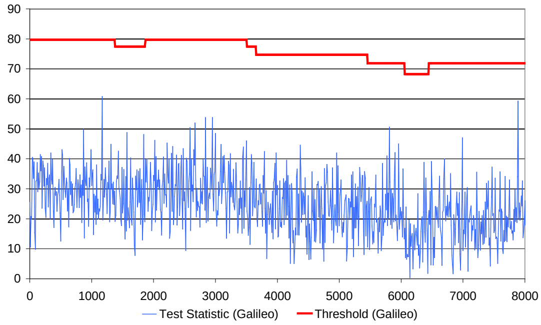


Fig. 16 Test statistic versus threshold for GPS group separation (i.e. Galileo only)

4. Conclusions

Carrier phased based integrity monitoring algorithms proposed in this paper successfully deal with the challenges identified and are able to monitor the accuracy at centimetre level. In particular, the new "difference test" integrity monitoring algorithm can detect slowly growing errors significantly earlier than conventional algorithms. Furthermore, the algorithm based on group separation concept exploiting failure mode data can significantly reduce the computation load required for failure exclusion. The combination of the algorithms developed with existing ones, should provide reliable integrity monitoring and quality control for seamless positioning in all conditions and environments.

Acknowledgements

The research presented in this paper was supported by the SPACE project. SPACE (Seamless Positioning in All Conditions and Environments) was an EPSRC-funded collaborative research project and came to an end during 2008. It began life as a Pinpoint Faraday flagship project but, following the evolution of Pinpoint into a Knowledge Transfer Network (KTN), operatesdwithin the Location and Timing KTN (http://www.locationktn.com). It was undertaken by a consortium of eleven UK university and industrial groups. The university groups were the CAA Institute of Satellite Navigation at the University of Leeds, the Department of Civil and Environmental Engineering at Imperial College London, the Department of Geomatic Engineering at University College London (UCL), and the Institute of Engineering Surveying and Space Geodesy at University of Nottingham, and the industrial partners were The Civil Aviation Authority, EADS Astrium, Leica Geosystems, Nottingham Scientific Ltd, Ordnance Survey, QinetiQ and Thales Research and Technology.

The primary aim of SPACE was to undertake the basic research needed to build a prototype GNSS-based positioning system that can deliver cm-level accuracy positioning everywhere and at all times. The main component research thrusts were multipath modelling and mitigation, quality control and assessment, measurement modelling and integration, and GNSS sensor design. A key output from SPACE was a the fully tested design of a prototype plug and play integrated positioning system that can be used as a test bed for current and future positioning components and algorithms.

References

Brown, A., and Mathews, B. (2006): ***A Robust GPS/INS Kinematic Integrity for Aircraft Landing.*** Proceedings of ION GNSS, Fort Worth, TX, September 2006, pp715-725.

Brown, G. (1992): ***A baseline GPS RAIM scheme and a note on the equivalence of three RAIM methods.*** Navigation: The journal of the institute of navigation.Vol.39,No.3, Fall 1992,pp. 301-316.

Brown, R.G., Chin,Y.G.(1998): ***GPS RAIM: calculation of thresholds and protection radius using Chi-Square methods-A geometric approach.*** Global Positioning System: Papers Published in NAVIGATION. Volume V, 1998, 155-178.

Chang,X.W., Paige,C.C. and Perepetchai, V.(2001): ***Integrity Methods Using Carrier Phase, Proceedings of International Symposium on Kinematic Systems in Geodesy.*** Geomatics and Navigation (KIS 2001), Banff, Alberta, Canada June 5-8, 2001, pp. 235-245.

De Jonge, P., and Tiberius, C., (1996): ***The LAMBDA method for integer ambiguity estimation: implementation aspects.*** Tech Rep. 12, Delft Geodetic Computing Centre, IGR-series.

Diesel, J., and Luu,S.,(1995): ***GPS/IRS AIME: Calculation of thresholds and protection radius using Chi-square methods.*** Proceedings of ION GPS 1995, Palm Springs, CA, 12-15, September 1995, pp1959-1964.

Ene, A., Blanch, J., Walter, T. (2006): ***Galileo-GPS RAIM for Vertical Guidance.*** Proceedings of ION NTM 2006, Monterey, CA, 18-20 January 2006.

Feng, S., Ochieng, W.Y. (2007a)**:** ***A Difference Test Method for Early Detection of Slowly Growing Errors in GNSS Positioning.*** The Journal of Navigation 60(3) 2007, pp427–442.

Feng S, Ochieng W Y., Hide, C., Moore, T. and Hill, C.(2007b): ***Carrier Phase Based Integrity Monitoring Algorithms for Integrated GPS/INS Systems.*** Proceedings of the European Navigation Conference, Geneva, Switzerland, 29-31, May, 2007.

Feng S, Ochieng W Y(2006): ***User Level Autonomous Integrity Monitoring for Seamless Positioning in All Conditions and Environments.*** Proceedings of the European Navigation Conference, Manchester, 7-10, May, 2006.

Gold, K.L., Brown, A.K., (2004), ***A Hybrid Integrity Solution for Precision Landing and Guidance.*** Proceedings of IEEE PLANs, Monterey, CA, April 2004, pp165-174.

Hide, C., Moore, T. and Hill, C., (2006): ***Development of a multi-sensor navigation filter for high accuracy positioning in all environments.*** Proceedings of ION GNSS, Fort Worth, TX, September 2006, pp1635-1644.

ICAO (2006): ***Annex 10 Aeronautical Telecommunications, Volume I Radio Navigation Aids (Sixth Edition).*** **23/11/2006.**

Lee, Y. C. (2004)**: *Investigation of Extending Receiver Autonomous Integrity Monitoring (RAIM) to Combined Use of Galileo and Modernized GPS.*** Proceedings of ION GNSS 2004, Long Beach, California, 21–24 September 2004.

Lee,Y.C., O'Laughlin,D.G. (1999): ***Performance analysis of a tightly coupled GPS/inertial system for two integrity monitoring methods.*** Proceedings of ION GPS 1999, Nashville, Tennessee, 14-17, September 1999, pp.1187-1197.

Michalson,W., Hua, H. (1995**), *GPS carrier-phase RAIM,*** Proceedings of ION GPS 95, Palm Springs, CA , September 1995, pp1975-1984.

Offer, C.R., Grove, P.D., Macaulay, A.A. et al, (2006): ***Use of Inertial Integration to Enhance Availability for Shipboard Relative GPS (SRGPS).*** Proceedings of ION GNSS, Fort Worth, TX, September 2006, pp726-736.

Pervan, B., Lawrencwe D.G., and Parkinson, R.W., (1998**): *Autonomous Fault Detection and Removal Using GPS Carrier Phase.*** IEEE Transactions on Aerospace and Electronic Systems, 34(3), pp897-906.

Pervan,B., Lawrencwe D.G., Cohen, C.E., and Parkinson, R.W., (1996): ***Parity Space Methods for Autonomous Fault Detection and Exclusion Using GPS Carrier Phase.*** Proceedings of IEEE PLANs, pp649-656.

Pervan, B., Chan, F., et al, (2003): ***Performance Analysis of Carrier Phase DGPS Navigation for Shipboard Landing of Aircraft.*** Navigation: Journal of the Institute of Navigation, 50(3), pp181-191.

RTCA/DO-229C (2001): ***Minimum operational performance standards for global positioning system/ wide area augmentation system airborne equipment.*** November, 2001.

Teunissen, P.J.G., (2005): ***Integer aperture bootstrapping: a new GNSS ambiguity estimator with controllable fail-rate.*** Journal of Geodesy, Volume 79, Numbers 6-7, August 2005.

Miami Redblade III: A GPS-aided Autonomous Lawnmower

G. Newstadt, K. Green, D. Anderson, M. Lang, Y. Morton, and J. McCollum
Miami University, Oxford, Ohio.

Abstract

This paper describes the technical aspects of the Redblade III, Miami University's third generation autonomous lawnmower. The Redblade III was created for entrance in the Institute of Navigation's 4th Annual Autonomous Lawnmower Competition by a team of undergraduate students majoring in electrical, computer, and mechanical engineering at Miami University. This paper details the five major subsystems of the lawnmower, including (1) the sensing system, (2) the control system, (3) the mechanical chassis system, (4) the safety system, and (5) the base monitoring and testing system. The paper discusses each aforementioned system in detail, along with providing cost analysis and conclusions.

Keywords: Autonomous vehicle, GPS, DGPS

1. Background Introduction

The Redblade III is the third-generation autonomous lawnmower designed at Miami University of Ohio for entrance in the Institute of Navigation's (ION) 4th Annual Autonomous Lawnmower Competition. Previous generations of the Redblade were entered in the ION competitions in 2004 [1] and 2005 [2]. Moreover, the fourth generation Redblade was recently entered in the 2008 competition, though it will not be discussed here.

The ION Autonomous Lawnmower Competition consisted of the design and testing of autonomous vehicles for mowing a lawn of known shape. The lawnmowers were required to have no remote controls outside of a wireless remote emergency stop capability. Moreover, no local installations (buried wires, poles) were allowed, except for a Global Positioning System (GPS) local base station. The competition's complexity has increased over the years by changing the shape of the field (rectangular to L-shaped) and including moving obstacles, among other changes.

The first generation Redblade [1] incorporated differential GPS (DGPS) and Hall-effect sensors for precise positioning, and a two level control system for path planning and error correction. However, the Redblade I base mower was modified from a commercial unit, and was both bulky and difficult to modify. The Redblade II [2] created a custom mechanical chassis to overcome these difficulties. Moreover, it replaced the Hall-effect sensors with much more effective and accurate optical encoders through the RobotEQ AX2550 system (see following sections for further description).

Lastly, the Redblade III was designed to improve on the previous generations in two important ways: (1) increased robustness through a redesigned DGPS system and the introduction of a digital compass; and (2) the ability to sense and react to moving obstacles. The rest of this paper outlines the design of the Redblade III in much greater detail, a relevant cost analysis, and conclusions.

2. Systems Overview

The design of the Miami Redblade III, is subdivided into five main systems: the sensing system, the control system, the monitoring and testing system, the safety system, and the base mower mechanical chassis system. Fig. 1 shows a flow diagram representing the relationships between these five bus-systems. Fig. 2 displays a picture of the final physical implementation of the lawnmower.

As stated above, Redblade III is an extension of previous autonomous lawnmowers at Miami University and it draws much of its design from its predecessors.

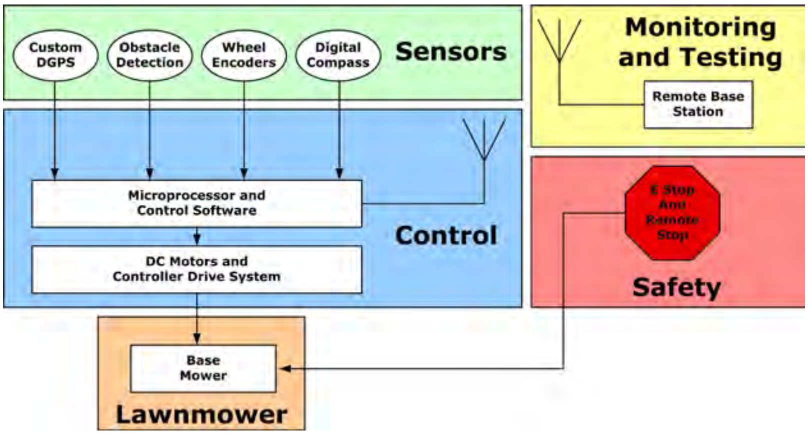


Fig. 1 Systems overview flow diagram

The current implementation employs the same mechanical chassis system and safety system of the Miami Redblade II. However, the current lawnmower has been upgraded with a modified DGPS system, new wheel encoders, and a more advanced control system. Furthermore, acoustic sensors and a laser ranging system have been added in order to supply obstacle detection capabilities.

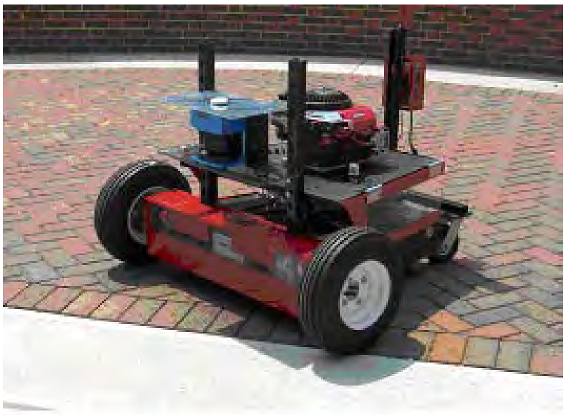


Fig. 2 Physical implementation of the Redblade III

3. Sensing System

The sensing system is comprised of three parts: a differential global positioning system receiver (DGPS), an electronic compass, wheel encoders, and acoustic sensors. The DGPS system consists of two NovAtel Superstar II GPS receivers [3], a wireless radio link, and custom carrier phase-based precision RTK position algorithms developed at Miami by the team. The Honeywell HRM3200 electronic compass [4] provides heading information during turning as well as ensuring the mower does not start to drift from its expected heading in between waypoints. The wheel encoders use a US Digital E7MS quadrature optical encoder [5] in order to determine the position and velocity of the lawnmower between DGPS solution updates. Each of these sensors will be discussed in greater detail in the following sections.

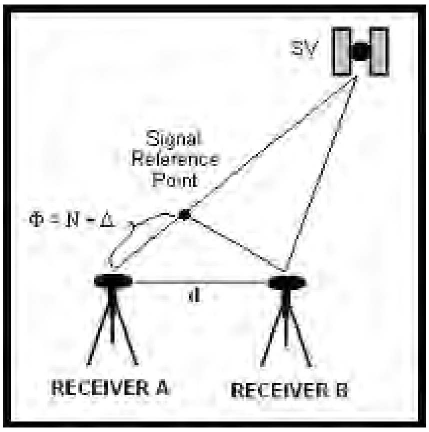


Fig. 3 Carrier phase integer ambiguity resolution

Custom DGPS

Navigation with GPS has become ubiquitous with the advent of personal GPS receivers in recent years. However, typical single frequency, civilian GPS receivers provide position accuracy only at the meter level [6]. The addition of another GPS receiver, on the other hand, allows the reduction of many correlated errors, including those due to the propagation through the ionosphere and troposphere, the satellite clock and orbit error, and the ephemeris error, provided that the baseline between the two receivers is not large. Our DGPS system is based on carrier phase measurements to provide accuracy at the centimeter level. Also, our system takes advantage of the fact that we initially know the exact relative positions of our receivers. This is done by precisely aligning the two receivers with a fixed distance between them. This allows our algorithms to quickly calculate the carrier phase integer ambiguities. Once these ambiguities have been calculated, one of the receivers is allowed to roam freely, and the relative positions are calculated using an iterative linear minimization algorithm. Fig. 4 displays a schematic representing the integer ambiguity resolution that is used in our code. For a detailed explanation of how the DGPS system works and all of the mathematics that is involved, see Appendix A.

The DGPS operates updates with a rate of 1 Hz. Since the lawnmower's allowed maximum speed is 10km/hour which implies that it can move about 3 m in one second. Such distance can greatly impact the quality of mowing and may have consequences in safety. It is important that other systems be employed to locate the vehicle in between the times of the DGPS updates. The subsequent section describes the wheel encoder system that is used for this purpose.

Wheel Encoders

The wheel encoders on Redblade III use US Digital E7MS quadrature optical encoders which are installed inside of the motors. Each encoder has two different signal channels which have phases that are 90 degrees apart. Each time the optical sensor detects a change, a pulse is sent to one of the signal channels, and a second pulse is sent 90 degrees offset from the first pulse. With this two channel configuration, detecting whether the wheel is moving forward or backwards becomes possible. Because the number of pulses there are per revolution of the wheel is known, dead-reckoning is used to compute the distance the mower has moved. This information is also applied to the encoders with a Proportional Integral Derivative (PID) control loop. In order to make both of the wheels turn at the exact same speed, an encoder module from RobotEQ [7] was purchased which was installed directly into the existing RobotEQ DC controller. The encoder module decodes the pulse train coming from the quadrature optical encoders and increments or decrements a counter register in the RobotEQ DC controller depending on if the wheel is going forward or in reverse.

This short-term dead-reckoning system not only fills the data gap between DGPS updates, it also provides redundant measurements to ensure the integrity of the DGPS. The DGPS is used to correct the errors that would accumulate if only wheel encoders were used to determine position.

Digital Compass

The Honeywell HMR3200 digital compass uses magneto-resistive sensors to determine heading information. The HMR3200 is a two-axis compass that is used to compute the azimuth angle of the lawnmower. The compass supplies data at rate of up to 15 Hz. The compass data is used primarily to orient the lawnmower turning rotations, though it can also serve to correct the path of the lawnmower if the heading diverges too much from the expected value.

Obstacle Detection

Two sensors were considered for obstacle detection and avoidance. The first is a SICK LMS200 Laser Range Finder (see Fig. 8). The LMS200 uses a laser to detect the distance an object is away from the unit, providing 180 degree visibility about a vertical axis and a 30 meter range. Furthermore, objects can be detected at a centimeter level accuracy.

The second sensor is a parallax acoustic sensor which uses the properties of sound to detect the distance of an object from the sensor. The acoustic sensor only has a range of 3 meters and accuracy much less than the laser ranging system described above. On the other hand, it is considerably cheaper and may be sufficient for the obstacle avoidance that our lawnmower requires.

Due to the overwhelming precision and accuracy of the laser ranging sensor, we decided to use the SICK LIDAR. However, we found that this system has the tendency to be tricked into thinking an obstacle is there when no obstacle exists (which can occur when sunlight is directly inputted in to the laser). Thus, future implementations may use acoustic sensors as redundant measurements.

4. Control System

The control algorithm is executed on a notebook computer that is mounted on the Redblade III. All of the various electronics, motor controllers, and sensors are connected to the computer using RS232 connections. The lawnmower incorporates a RobotEQ DC Motor Controller that is also controlled by the computer. The RobotEQ DC Motor Controller has a built-in PID control loop that enables the two separate motors to move concurrently and at the same speed. The wheel encoders are also connected directly to the RobotEQ, providing the fastest information to the PID controller.

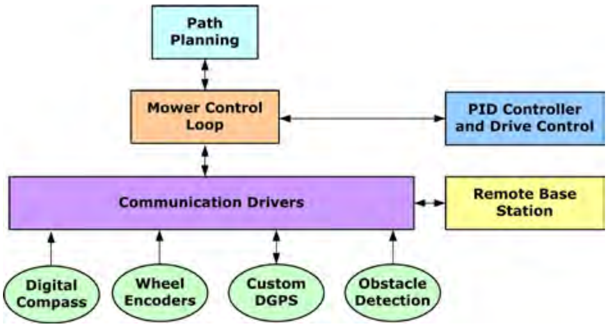


Fig. 4 Control systems integration flow diagram

Overview and system integration

The control software consists of four major components: (1) High-level path planning and control; (2) a control loop to determine and correct the current lawnmower position with regard to the path-planning, sensor inputs, and obstacle detection; (3) low-level communication interfacing between the control loop and the sensors, the actuators, and the remote base station; and (4) a PID controller to direct the lawnmower while it is moving. Fig. 4 shows a flow diagram of the integration of these systems.

The control software provides the option to control the path planning through the remote base station for testing and monitoring purposes. Furthermore, several exterior utilities were created to complete such tasks as forecasting satellite availability.

All of the software is written in Java. An object-oriented approach was implemented to provide the most flexibility for the project. Extensive class libraries were created for the systems described above, and a detailed model description of these libraries is available upon request.

Control Algorithm

The control algorithm is shown as a flow diagram in Fig. 5. The algorithm is composed of four main components: (1) System initialization; (2) path planning and control, including the ability to dynamically change the planned path based on obstacle detection and/or discovered errors in the path travelled by the lawnmower; (3) orientation change; (4) position change; and (5) obstacle detection.

It is important to note several things about the control algorithm. First, while the lawnmower is moving, all position estimates are computed using the information supplied by the wheel encoders. The compass will provide heading information to control the turning angles at the desired location. Furthermore, the PID controller uses the wheel encoder data to dynamically control the drive system during this time.

Second, the DGPS system is used to calculate precise locations once the lawnmower has come to a stop, which occurs when the lawnmower has reached its desired location or has encountered an obstacle. However, the DGPS calculated position may not line up exactly with the desired location of path planning, and the detected obstacle may make it impossible to travel to the correct endpoint. At this point, the path planning's dynamic capabilities allow the lawnmower to update its next desired position based on the decision to correct any error in the path already travelled or to avoid an obstacle.

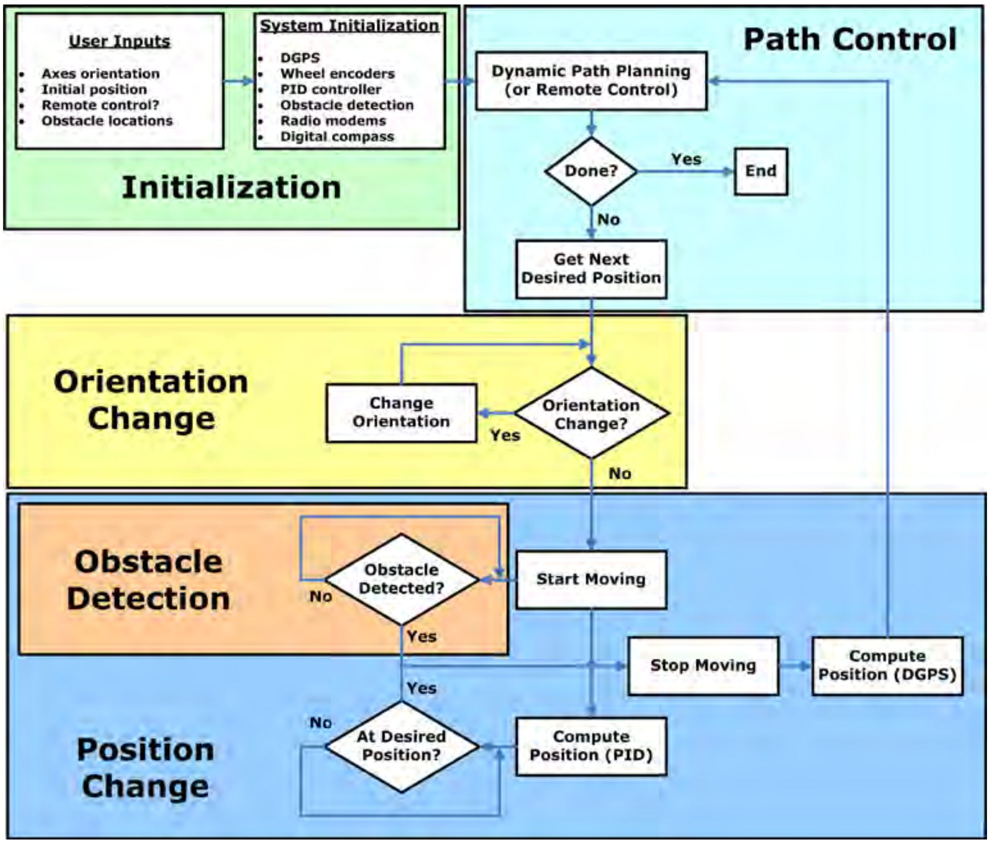


Fig. 5 Control algorithm flow diagram

Third, while testing has given us confidence that our systems work as designed, we have built in robustness checks based on the redundant data given by our multiple sensors. Furthermore, the DGPS system has the ability to re-initialize itself if it has determined that it is no longer functioning in a valid state. This algorithm is described in the subsequent section.

DGPS Control

The DGPS is normally a well-functioning system. However, occasionally the system will cease to operate in a valid state, such as when it ceases to track a minimum of four satellites. This can occur if the signals from the tracked satellites are blocked in some fashion. For this reason, it is important that mission planning be done before the lawnmower is actually operated, and the aforementioned satellite availability forecasting software was designed for exactly this purpose.

Nevertheless, with the possibility of invalid position data being generated by the DGPS, an algorithm to re-initialize the system was developed for the sake of robustness and reliability. The flow diagram of this algorithm is shown in Fig. 6. It is important to note that the DGPS determines whether it is in a valid state within the software package. Considerations for validity include the number of tracked satellites, limits on the calculated positions with respect to previously calculated positions (the lawnmower can only move so far in a set period of time), and consensus with the redundant data given by other sensors. Additionally, when being re-initialized, the DGPS has to assume that the positions given by the other sensors are completely accurate. While this may introduce some error to the system as a whole, the DGPS is integral to a fully-functioning autonomous lawnmower, so the error is tolerated.

Path Control Algorithm

The path control is divided into two major components: (1) path planning; and (2) decision making based on external sensors. This algorithm is shown in a flow diagram in Fig. 7. The path planning is computed initially before the lawnmower begins moving and outputs a set of waypoints for the lawnmower to follow. At each waypoint, the lawnmower updates its position through the DGPS, changes its orientation, checks its heading, or does some combination of these actions.

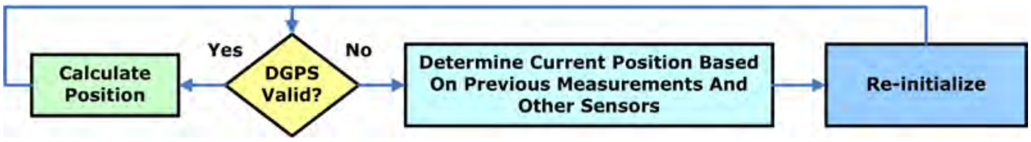

Calculate Position
Yes
DGPS Valid?
No
Determine Current Position Based On Previous Measurements And Other Sensors
Re-initialize

Fig. 6 DGPS control flow diagram

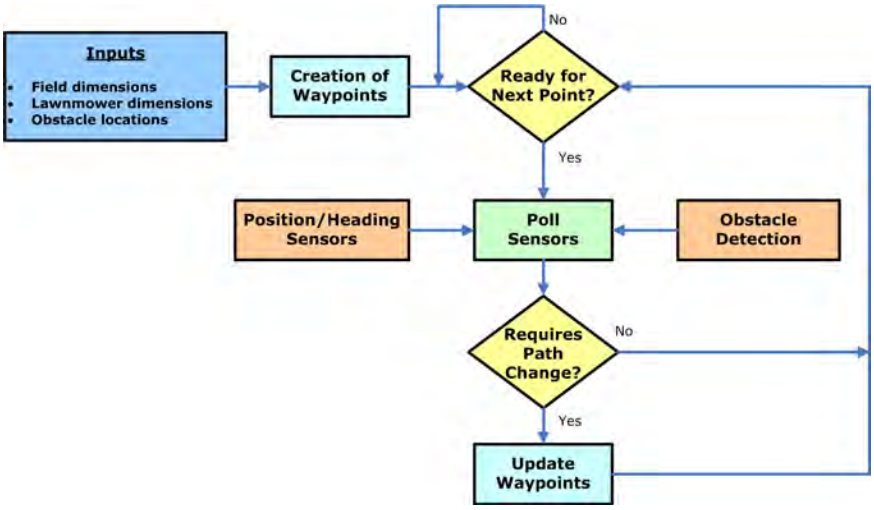


Fig. 7 Path control flow diagram

The path planning is computed based on a set of initial parameters. These parameters include the field's dimensions (assuming a rectangular geometry), the obstacle size and locations, and the lawnmower's dimensions (such as width, length, and cutting blade length). The waypoints are generated in such a way to insure that the lawnmower never leaves the boundary while moving or turning. Furthermore, the turns are constructed in such a way to never place any part of the mower outside of the boundary. Additionally, static obstacle avoidance is pre-computed to make arbitrary radial turns that will allow for smooth motion around each obstacle. These turns are done to ensure mowing in a safe zone as well as to be aesthetically pleasing. Fig. 8 shows a graphical output of the initial computed waypoints given two obstacles with different sizes.

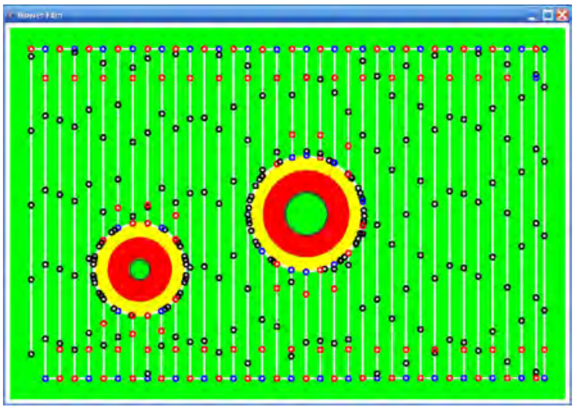


Fig. 8 Waypoint configuration for two obstacles with different sizes

The path control also incorporates decision making based on external sensors. This includes updating the path waypoints when an obstacle is encountered or when position sensors (location and heading) indicate that the lawnmower is off-target past a certain threshold.

5. Drive and Power System

The control system described in the previous section is critical to the functioning of the Redblade III, but without the drive and power system, it would be completely useless. The Redblade III incorporates a drive system with two Power Chair (NPC) model T64 24-Volt DC motors and a hybrid power system consisting of rechargeable batteries and a gas engine. The individual drive and power systems are described in the following sections.

Drive system

The drive system consists of two Power Chair (NPC) model T64 24-Volt DC motors. The DC motors are voltage-controlled with a low RPM-torque of approximately 300 in-lbs. Furthermore, the motors are equipped with a 20:1 gear ratio to give suitable RPM ranges for operation. The DC motor and wheel couplings are pictured in Fig. 9.

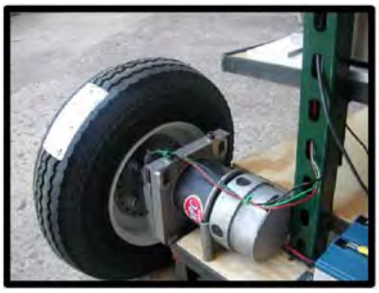


Fig. 9 DC motor and wheel coupling implementation

Power system

The power system incorporates both rechargeable batteries and a gas engine. The Redblade III employs two Power-Sonic sealed lead acid (SLA) batteries. The low-cycle batteries output 12 Volts and 7 amp-hours, and are connected in series to provide 24 V to the DC motors and the various on-board electronics. The batteries are rechargeable through ordinary AC power outlets while the lawnmower is stationary. Currently, our design does not incorporate any way to charge the batteries while the lawnmower is operating, though future work includes integrating an alternator for this purpose.

The Redblade III also utilizes a 5.5 horsepower gas engine in order to provide sufficient rotational energy to the cutting blade. Fig. 10 highlights the power system on board of the lawnmower.

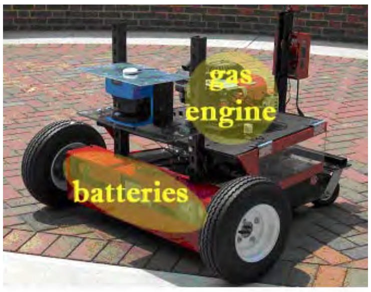


Fig. 10 Power system on the Redblade III

Wiring diagram

The wiring diagram of the drive and power system is shown in Fig. 11.

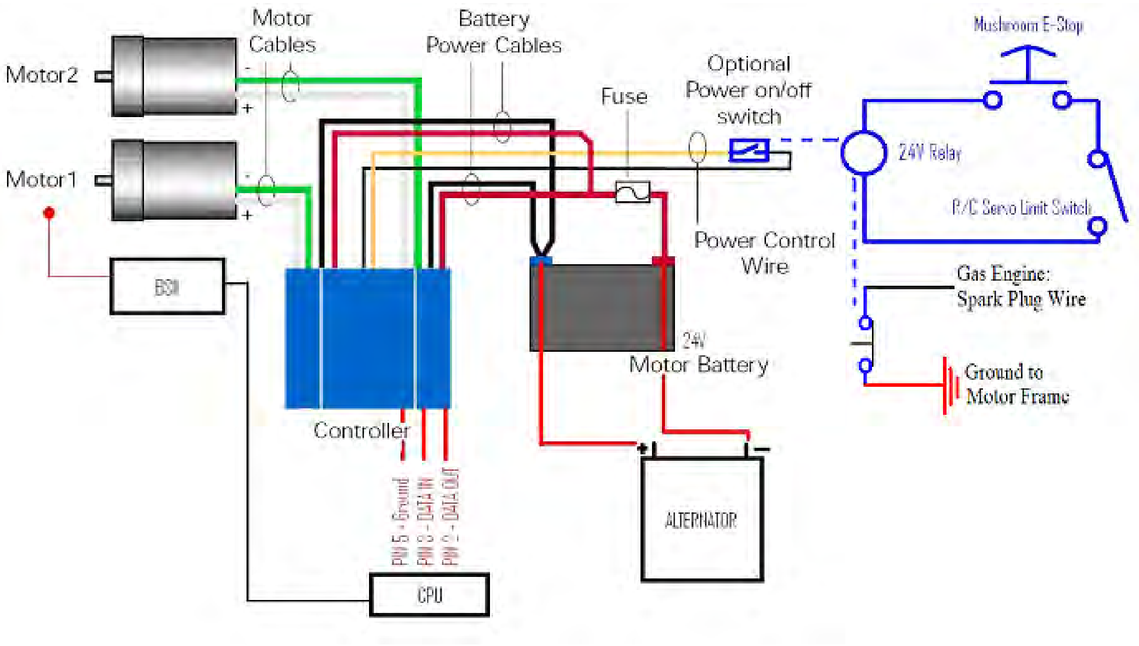


Fig. 11. Wiring diagram

6. Mechanical Chassis System

The mechanical chassis system was designed to optimally integrate all of the previously discussed systems in a physical manner. The lawnmower is lightweight, yet robust. It is framed with angle iron, and the mounting surfaces are shielded with high strength steel sheeting with plywood covering the steel. Furthermore, a shelving system was incorporated to offer the maximum flexibility to our layout and construction.

The top shelf (see Fig. 12) houses the gas engine. The bottom shelf (see Fig. 13) holds most of the electronics, including the notebook computer, the batteries, the RobotEQ controller, the GPS receiver (and radio modem), and the power circuitry. Specialized mounts were created for the laser ranging system, the safety switch, the digital compass, and the GPS antenna (see Fig. 14).

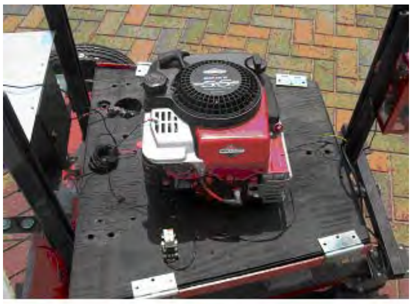


Fig. 12 Top shelf (gas engine)

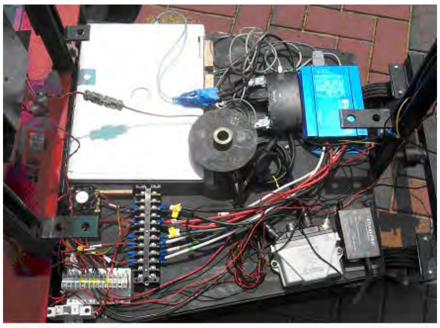


Fig. 13 Bottom shelf (electronics, batteries, etc.)

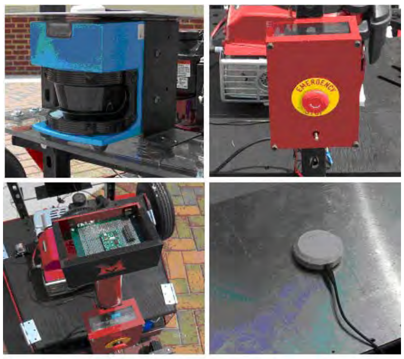


Fig. 14 Specialized mounts: laser ranging system (top left), safety switch (top right), digital compass (bottom left), and GPS antenna (bottom right).

Two 6" pneumatic caster wheels with 4" x4" mounting plates were custom-designed for the lawnmower. The pneumatic nature assists in the handling of rocky and unstable terrains, such as may be the case with mowing field. Additionally, the casters were mounted to a shaft in the front to allow the caster assembly to pivot vertically. This allows either front wheel to encounter a ditch or imperfection in the field without causing the rear wheels to lift up. This design ensures that the base of the lawnmower remains at a relatively constant height and gives the lawnmower the ability to always propel itself out of a hole. Fig. 15 shows the implementation of the caster wheels. The rear wheels drive the vehicle with diameters of 16.5".

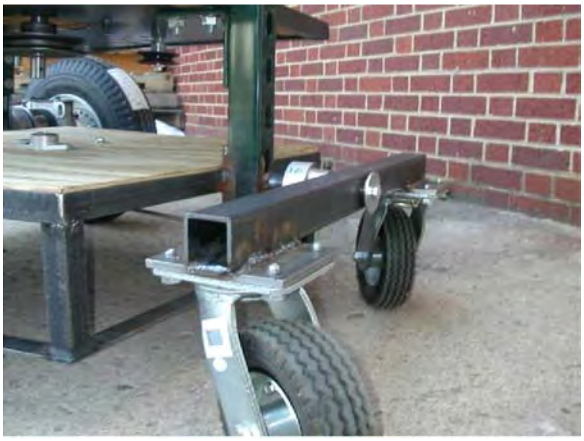


Fig. 15 Caster wheels implementation

A unique shaft coupling design was created to link the gas engine to the alternator shaft and the cutting blade. This provides the ability to disengage the blade while still running the alternator, which was very desirable for testing purposes. Fig. 16 displays the shaft coupling mechanism.

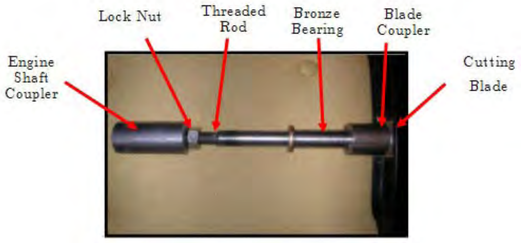


Fig. 16 Shaft coupling mechanism

7. Safety System

With a large vehicle attached with a cutting blade that could cause considerable damage, it is of utmost importance that a reliable safety system be implemented. For the Redblade III, an on-board emergency stop and a remote-controlled emergency stop provided for this purpose. The emergency stop system allows the user to stop all motion on the lawnmower (e.g., the DC motors and the gas engine). Fig. 17 shows the emergency stop circuit. A 24 V relay controls this circuit, which can be broken by a normally closed emergency stop button that is easily accessible from the rear of the lawnmower. A normally open limit switch also can cause the power circuit to be broken. The limit switch is held closed by an RC servo that is kept in tension via a spring mounted to the control panel. Thus, the user can easily open the limit switch (causing the power circuit to be broken) by releasing the RC control trigger. Furthermore, since the limit switch is normally open, if the RC controller is dropped or loses power, the emergency stop will be activated, creating a desired fail-safe mode of operation.

The RobotEQ motor controller has an optional on/off switch controlled by two wires connected through a normally closed port controlled by the 24 V relay. Thus, if the relay loses power (i.e., the power circuit has been broken) then the RobotEQ will also lose power.

Stopping the motion of the gas engine requires that the spark plug be grounded to the motor frame. The 24 V relay is therefore connected in series with the spark plug, causing the gas engine to lose power when the 24 V relay loses power.

8. Base Station Monitoring and Testing Station

A base station for remote monitoring and testing was developed to accompany the Redblade III. The base station is comprised of a PC with wireless communication capabilities, a custom-designed user interface, remote control, and data logging programs. The remote monitoring and testing software was written in Java.

9. Conclusions

The Redblade III is Miami University's third generation autonomous lawnmower. It has incorporated many changes, including the addition of robust custom DGPS, advanced control algorithms, wheel encoder sensors, obstacle detection capabilities, and an updated mechanical chassis.

Further improvements could include replacing the onboard notebook computer with a dedicated microprocessor, as well as using an inertial momentum unit (IMU) to replace the noisier digital compass. Lastly, the use of multiple sensors may lead us to use more advanced, adaptive processing for control. At the very least, we could employ Kalman or particle filtering to provide optimal (or near-optimal) control.

Overall, the Redblade III is much more robust and reliable than in previous generations, though it still offers much of the same flexibility and ability for improvement that was seen in the Redblade II. Although there is still room for significant improvement, we are pleased with the progress of the lawnmower and believe that the autonomous lawnmowers may be an achievable consumer goal in the near future.

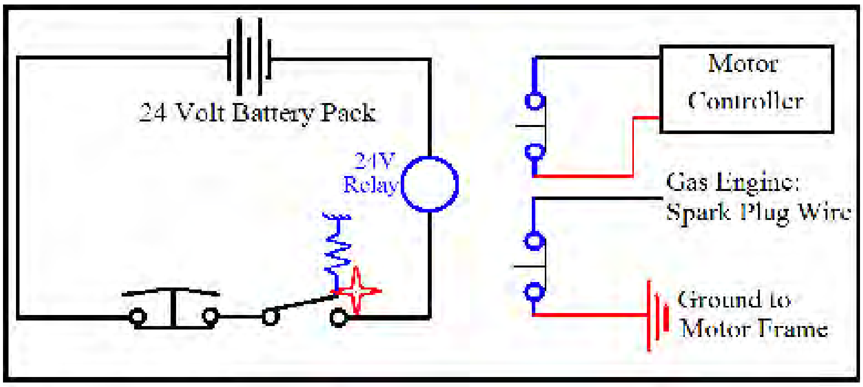


Fig. 17 Emergency stop circuits for the 24 V relay (left), RobotEQ controller (top right), and gas engine (bottom right)

Appendix: DGPS algorithm description

The integer ambiguities associated with the carrier phase are integral to the precise positioning of the user. This methodology of ambiguity resolution takes into account the fact that we originally know the *exact* distance between our two receivers at the initial time. Fig. 3 shows the two-dimensional arrangement of a satellite and our two receivers. When we know the distance between the USER and the REFERENCE (12 inches) and the related pseudoranges, the ambiguity, N, is easily solved through some basic geometry.

However, due to errors in the atmosphere (ionospheric, tropospheric delays) and satellite clock errors, we would not expect reliable ambiguity calculations by using just one satellite. Instead, we use double-differencing techniques to remove these correlated errors. The formula we use to calculate the ambiguities is given by:

$$N_{ur}^{i,1} = \frac{R_{ur}^{i,1}}{\lambda} - \phi_{ur}^{i,1}.$$

where R refers to the range in meters, and ϕ is the carrier phase in radians. Also, the subscripts refer to the USER (u) and the REFERENCE (r) receivers, the superscripts refer to the BASE satellite (1) and the other satellites (i), and the notation in the formula is defined as:

$$(*)_{ur}^{i,1} = \left(\left((*)_u^i - (*)_u^1\right) - \left((*)_r^i - (*)_r^1\right)\right).$$

Furthermore, since we already know the original orientation of our USER and REFERENCE receivers, and the carrier phases are provided by the receivers themselves, we only have 1 equation with 1 unknown, and our ambiguity resolution is complete. However, it is important not only to resolve the ambiguities, but to also to consistently calculate the *same* ambiguities over a period of time. Therefore, the code requires that each ambiguity must be calculated the same way 20 times in a row before allowing the USER to roam.

The range equation with regard to the carrier phase is given by

$$\lambda\phi_r^i = \lambda\hat{\phi}_r^i + C^i + I_r^i + T_r^i$$

where the latter terms refer to the satellite clock error, ionospheric delay, and tropospheric delay, respectively. However, from equation (1), we know we can solve for the USER position by knowing the relationship:

$$R_u^{i,1} - R_r^{i,1} = R_{ur}^{i,1} = \lambda\phi_{ur}^{i,1} + \lambda N_{ur}^{i,1}$$

Furthermore, if we use a first-order Taylor expansion, and expand it to three dimensions (xyz), we will get the matrix equation:

$$A_j^i * D = L^i$$

for i = 2,..,N and j = 1,2,3 and

$$A_1^i = \frac{X^i - X_u(0)}{R_u^i(0)} - \frac{X^1 - X_u(0)}{R_u^1(0)}$$

$$A_2^i = \frac{Y^i - Y_u(0)}{R_u^i(0)} - \frac{Y^1 - Y_u(0)}{R_u^1(0)}$$

$$A_3^i = \frac{Z^i - Z_u(0)}{R_u^i(0)} - \frac{Z^1 - Z_u(0)}{R_u^1(0)}$$

$$D = \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{bmatrix}$$

$$L^i = \lambda\phi_{ur}^{i,1} + \lambda N_{ur}^{i,1} + \left(R_r^i(0) - R_r^1(0)\right) - \left(R_u^i(0) - R_u^1(0)\right)$$

where variables with subscripts refer to measurements from the receivers, while variables with superscripts refer to measurements from the satellites. Also, this system of equations can be solved with a least-squares solution to get:

$$D = \left(A'Q^{-1}A\right)^{-1}\left(A'Q^{-1}\right)L$$

where Q is the sample covariance matrix. The user position is then given by:

$$R_u = \begin{bmatrix} X_u \\ Y_u \\ Z_u \end{bmatrix} = R_u(0) - D = \begin{bmatrix} X_u(0) - \Delta X \\ Y_u(0) - \Delta Y \\ Z_u(0) - \Delta Z \end{bmatrix}$$

Lastly, this process of calculating the position is done iteratively until the delta matrix, D, becomes approximately zero (less than 1e-9).

References

[1] McNally B., Stutzman M., Koranda C., Mantz C., Macsek J., Miller S., Walker A., Morton J., Campbell S., Leonard J (2004) ***The Miami Redblade: Technical Report***, Institute of Navigation Autonomous Lawnmower Competition.

[2] French M., Russler J., Smith J., Smith L., Walters T., Morton J., Campbell S., Leonard J. (2005) ***The Miami Redblade II: Technical Report***, Institute of Navigation Autonomous Lawnmower Competition.

[3] NovAtel Superstar II GPS receiver circuit board, http://www.novatel.com/products/superstar.htm.

[4] Honeywell HMR3200/HMR3300 digital compass, http://www.ssec.honeywell.com/magnetic/datasheets/hmr32003300.pdf.

[5] Metal optical kit encoder – e7m, http://www.usdigital.com/products/e7m/.

[6] Misra P. and Enge P. (2005) Global Positioning Systems: Signals, Measurement, and Performance. Ganga-Jamuna Press: 147-282.

[7] Roboteq: Ax2550, http://www.roboteq.com/ax2550-folder.html.

A Conceptual Framework for Server-Based GNSS Operations

Samsung Lim and Chris Rizos
University of New South Wales, Sydney, Australia.

Abstract

The diversification of Global Navigation Satellite Systems (e.g. the current and modernized GPS, the revitalized GLONASS, the planned Galileo and Compass), is an opportunity for engineers, surveyors and geodesists because of expected improvements in positioning accuracy, operational flexibility, redundancy, and quality assurance. Recent research activities include new algorithms for multiple frequency ambiguity resolution, software-based receivers for re-configurability, network-wide corrections for utilising redundancy, reversed real-time kinematic schemes for quality/accuracy improvement, and a wide range of rover-side applications. This paper discusses the integration of these "pieces" of work into a new framework and facilitates information and communication technologies in order to derive benefits from network infrastructure such as continuously operating reference stations and local/regional GPS networks. Operational models are proposed for precise point positioning and real-time kinematic services including "near real-time" applications, which require an optimal design to balance the computational overhead with data communication latency. The proposed framework is designed to be a comprehensive, server-based, and thin-client platform. It provides end-users with "out-of-the-box" services. End-users should be able to obtain extensive GNSS capabilities and high productivity without conventional constraints such as an expensive set of receivers, proprietary data formats, user-installed carrier phase processing software, incomplete interoperability, limited communication links, etc. The framework also adopts up-to-date database technologies and web technologies that enable servers to perform data management and spatial analysis, while end-users are able to syndicate data and create their own business models. The framework has been applied to Sydney Network (SydNET), a network of continuously operating reference stations located in Sydney, Australia. It is expected that the new framework will be versatile enough to cope with a diverse range of user performance requirements and the operational requirements for communications and positioning computations.

Keywords: GPS, GNSS, Server-based, Framework.

1. Introduction

By definition, a server-based thin-client real-time kinematic (RTK) requires a designated server to compute a rover's coordinates in the required reference system by taking advantage of existing GNSS reference network infrastructure, instead of broadcasting corrections or data to users and placing the onus of obtaining a final solution on clients and their equipment. Final (position) solutions for all real-time (logged) users could be simply computed as a by-product of the continuous network processes – all the time satisfying the quality and integrity criteria implemented at the network administrator level (Rizos & Cranenbroeck, 2006).

Note that improved accuracy and reliability of the user coordinates can be expected if GNSS data is processed in the network mode (e.g. as implemented in network-RTK schemes), rather than as individual baselines as is the case of standard RTK-type techniques (i.e. single-base RTK). In addition, precise ultra-rapid ephemerides produced by International GNSS Service (IGS) can be used in network-RTK instead of the broadcast ephemeris. For example, network-RTK software "SNAPper" developed by the School of Surveying and Spatial Information Systems at the University of New South Wales, Australia, is able to generate real-time network corrections, based on IGS ultra rapid orbits. SNAPper is functionally equivalent to Trimble® GPSNet/VRS™ (Trimble, 2007) or Leica® SpiderNet™ (Leica, 2007). After all, there exist already a number of web-based services for the generation of coordinates via the post-processing of data submitted to a server by the client user. What is suggested in this paper is therefore to extend this capability to real-time data processing.

A server-based approach reverses the data flow in conventional RTK by requiring the user to transmit their data to the main server – sometimes also referred to as "reverse RTK" or "remote RTK" or "inverted RTK". Note that there is still the need for two-way communications between the client (field user) and the server (computer centre). The server software can select the optimal combination of continuously operating reference stations (CORS), and compute the best possible position solution before returning the result to the field user. The user then receives not only raw coordinates, but also a value-added product such as positioning quality indicators. Service providers can now exercise control over the generated products and, as a result, place a true commercial value on the service.

In addition, the user does not have to learn complicated GNSS surveying techniques or software. Safeguards, and thus integrity, can also be easily implemented into such a service. For example, if the number of satellites is too low, the geometry is unfavourable, or the multipath effects too detrimental, a message can be sent back to the user warning them that the provided solution is not optimal and that it may not meet their specifications. With the critical processes of legal traceability and integrity looming on the horizon for positioning services, such a total quality assured coordinate service may become increasingly attractive. For example, Nippon GPS Solution has implemented an inverted RTK service in Japan (Kanzaki, 2006), and is marketing their service by promoting the quality assurance aspects of server-based RTK processing.

2. Base Station Selection

Although it has been proven that network-RTK is superior to single-base RTK (Rizos & Han, 2003), there is a need to switch off network-RTK and turn on single-base RTK in certain cases, such as when there is a communication fault between a reference station and a distributed server, or during maintenance of a base and/or a server, or in the event there is low quality data from a reference receiver (for whatever reason). The fault detection module of a distributed server detects such faults and then queries the main database server to obtain the information on the most suitable reference station for single-base RTK. The main database server stores the geometry and topology information for this purpose, and the local database server stores the data quality information for the reference stations. This database approach is faster and more reliable than a simple selection on the basis of which reference station is nearest to the rover.

In this paper, the state of New South Wales (NSW), in Australia, is used as an example to illustrate the reference station selection logic. The basic assumption is that 220 cities and towns across NSW are sampled as reference receiver sites. Distances from these sites are graphically shown in Figure 1. Yellow regions represent the distance within 10km of a reference station, so that single-base RTK is suitable. Green areas and yellow-green areas are within 50km from a reference station where server-based RTK is ideal. Light blue to green areas are within 100km from a reference station and therefore server-based network-RTK is preferred. There is a significant amount of blue coloured area, outside the 100km radius of a reference station. These rural areas are expected to suffer a lower accuracy. This distance distribution can also be interpreted as a proportional error distribution of RTK as far as the geometric correlation is concerned.

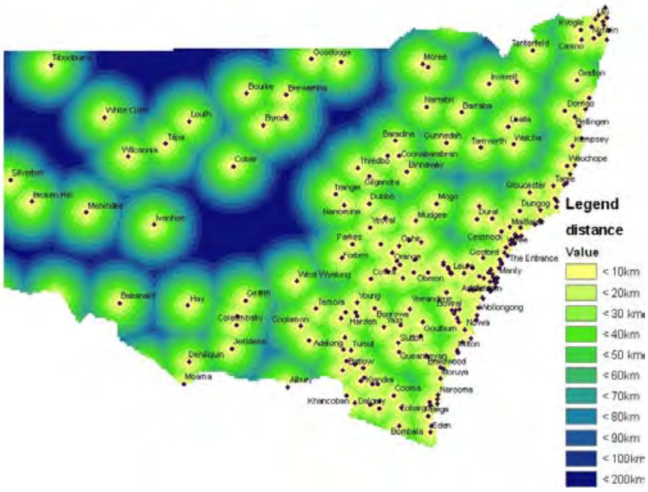


Fig. 1 Distance distribution of the assumed NSW network

Rather than calculating the distance from a rover on an *ad hoc* basis for the purpose of selecting the nearest reference station, Voronoi polygons in Figure 2 would be convenient. Voronoi polygons are obtained from the locations of the CORS so that each polygon contains only one reference station which is the nearest site from any location in the polygon. Therefore any rover within the polygon can perform single-base RTK with that particular reference station. Voronoi polygons can be obtained from Delaunay triangles and vice versa.

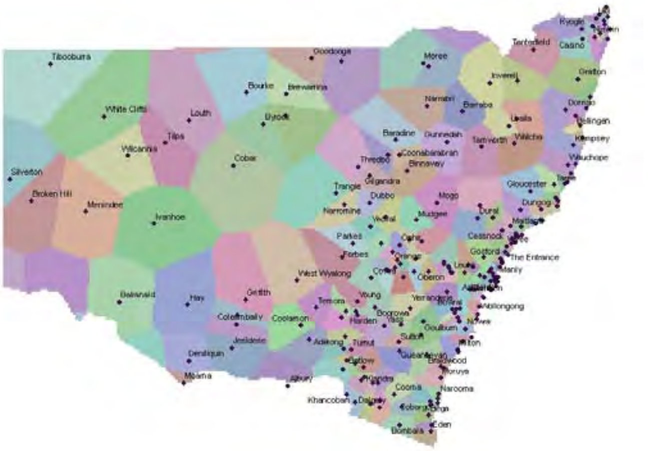


Fig. 2 Voronoi polygons of the assumed NSW network

Voronoi polygons must be stored in the main database server (see Section 3.4). Storing the geometry and topology of Voronoi polygons in the database is now feasible because of the spatial extensions of a Database Management System (DBMS). The benefit of storing such information in a database is that a query can be made as to whether a rover is located in the interior, or on the boundary, or exterior to a polygon, without the cost of computing the spatial relationship based on the coordinate information.

As for network-RTK, a combination of three or more reference stations can be selected to generate network corrections. However, three stations are sufficient and efficient in most cases. SNAPper uses three stations all the time, while GPSNet™ and SpiderNet™ have an option to choose more than three stations. To determine three reference stations, i.e. to form triangles, a Delaunay triangulation is more effective than any other triangulation. Delaunay triangles from locations of the CORS assure the condition that no reference station is located inside the circum-circle of any triangles. That is, Delaunay triangles maximise the minimum angle of triangles so that thin sliver triangles can be avoided, and therefore maximise the benefit of interpolating geometric correlations. Delaunay triangles and Voronoi polygons are geometrically paired off, i.e. Delaunay triangles can be defined by Voronoi polygons and vice versa.

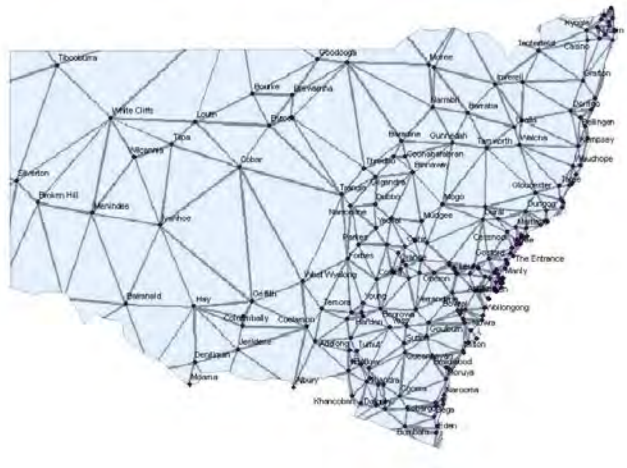


Fig 3 Delaunay triangles of the assumed NSW network

Figure 3 illustrates the Delaunay triangles for the assumed NSW CORS network. Again, the geometry and topology of Delaunay triangles can be stored in the main database server so that it is possible to identify if a rover is located in the interior, or on the boundary, or exterior to the triangle. The area of each triangle can be interpreted as a proportional error distribution of network-RTK unless error sources other than the geometric correlation are considered.

3. System Architecture

The system architecture of the proposed server-based thin-client RTK is indicated in Figure 4. The basic assumption is that distributed-computing is necessary to cope with simultaneous requests from hundreds of clients. Distributed-computing allows computers to efficiently communicate and individually process data, which is different from networked-computing.

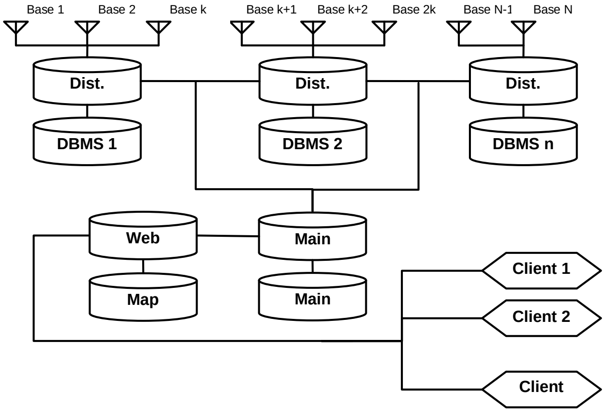


Fig. 4 System architecture of a distributed-computing based RTK service

4. Distributed Server

A distributed server receives data streams from k reference stations ($k \geq 3$) where k reference stations form as many Delaunay triangles as possible. Exclusive sets of reference stations are allocated to distributed servers, so that the last distributed server may have less than k reference stations unless the number of reference stations is divisible by k. For example, if a CORS network consists of Stations 1, 2, …, 6 and has 4 Delaunay triangles as depicted in Figure 5, then Stations must be allocated as described in Table 1.

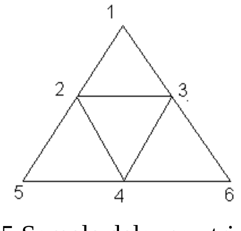


Fig. 5 Sample delaunay triangles

Table 1. Sample allocation of reference stations

k	Distributed Server 1	Distributed Server 2
3	1, 2, 3	4, 5, 6
4	1, 2, 3, 4	5, 6
5	1, 2, 3, 4, 5	6
6	1, 2, 3, 4, 5, 6	

Table 2. Sample allocation of Delaunay triangles

k	Distributed Server 1	Distributed Server 2
3	Δ1-2-3 Δ2-3-(4)*	Δ(2)-4-5 Δ(3)-4-6
4	Δ1-2-3 Δ2-3-4	Δ(2)-(4)-5 Δ(3)-(4)-6
5	Δ1-2-3 Δ2-3-4 Δ2-4-5	Δ(3)-(4)-6
6	Δ1-2-3 Δ2-3-4 Δ2-4-5 Δ3-4-6	

*Reference stations in brackets are received from other distributed servers.

If a distributed server does not include a reference station that forms a Delaunay triangle, then the distributed server must receive data streams from other distributed servers. Table 2 illustrates the situation. For example, in case of $k = 3$, Distributed Server 1 receives data streams directly from Stations 1, 2, and 3; and receives data streams indirectly from Station 4 via Distributed Server 2. Then server-side network-RTK in the area of two Delaunay triangles Δ1-2-3 and Δ2-3-4 can be serviced by Distributed Server 1. On the other hand, Distributed Server 2 receives data streams directly from Stations 4, 5, and 6; and receives data streams indirectly from Stations 2 and 3 via Distributed Server 1. As a result, all distributed servers can service server-side network-RTK for all possible Delaunay triangles. Note that the case of $k = 3$ is the most well-balanced distribution in terms of data transmission and reception or the computational workload.

Networked Transport of RTCM via Internet Protocol (NTRIP) is the chosen protocol for GNSS data transfer in this system. Each distributed server is equipped with an NTRIP caster so that the distributed server can broadcast data streams received from the allocated reference stations. Although the web server is the first contact point for all clients, the main server has the authority to permit advanced or privileged client access to a distributed server. Such a client can obtain direct data streams and perform traditional network-RTK (or single-base RTK), i.e. their own network algorithm can be applied to the data streams. It should be noted that an NTRIP client can be any Hypertext Transfer Protocol (HTTP) retrieval program, e.g. PHP, JavaScript, cURL, Wget, etc.

A normal client is expected to request the web server (and the web server to request the main server) to perform a server-based RTK computation. The main server determines if either single-base RTK or network-RTK is appropriate, depending on the rover requirement and the data quality of the reference stations. Then the main server assigns a distributed server to the client so that the distributed server performs a server-based RTK solution. The distributed server also parses "raw" data streams via NTRIP and inserts the data into the database of the local database server.

As for server-based network-RTK, the distributed server obtains precise ephemerides from the main database server and generates network corrections by interpolating residuals (or an extrapolation if the rover is not located within any of the triangles). Researchers have proposed a range of interpolation techniques: Linear Combination Model (LCM), Distance-Based Linear Interpolation Method (DBLIM), Linear Interpolation Method (LIM), Low-order Surface Model (LSM), and Least Squares Collocation Method (LSCM). Dai (2002) and Fotopoulus & Cannon (2001) reviewed these algorithms. Basically, these algorithms assume that errors are all spatially correlated, and therefore they are simply a variant of inverse distance weight (IDW) interpolation.

In order to improve the interpolation algorithm and to take into account the non-spatial correlation, the Kriging technique is a worthwhile alternative. Kriging is similar to regression analysis. Kriging is based on the assumption that the parameters being interpolated should minimise the estimation variance by applying an empirical covariance model. Kriging is only useful when the number of reference stations is large (≥ 50) so that the spatial correlation between reference stations can represent the empirical covariance model. Therefore the main server is ideal for computing Kriging parameters. Then each distributed server can use the parameters to perform Kriging for a client. This guarantees more uniformly distributed accuracy within the network than the traditional IDW interpolation because the parameters are obtained from network-wide observations. There are many types of Kriging, however, Ordinary Kriging is suitable for this application since there are enough observations to estimate the variogram, and for the state-wide area coverage of the NSW CORS network, the mean residuals apparently show unknown constant trend.
In summary, a distributed server should be able to perform the set of functions listed in Table 3.

Table 3. Distributed server functions

Function	Task
NTRIPClient1	To receive data streams from multiple reference stations via NTRIP
NTRIPClient2	To receive data streams from another distributed servers via NTRIP
NTRIPCaster	To broadcast multiple data streams via NTRIP

FaultDetection	To detect system faults such as a communication fault between a reference station and the distributed server
DataParser	To parse multiple data streams (Raw, RTCM, RT-IGS)
InsertDB	To insert the data into databases
ServerRTK	To perform server-based single-base RTK
ServerNetworkRTK	To perform server-based network-RTK
IDWInterpolator	To interpolate double differenced residuals (LCM, DBLIM, LIM, LSM, LSCM)
Kriging	To interpolate double differenced residuals (Ordinary Kriging)
NetworkAR	Network ambiguity resolution
NetworkResidual	To compute double-differenced residuals

5. Local Database Server

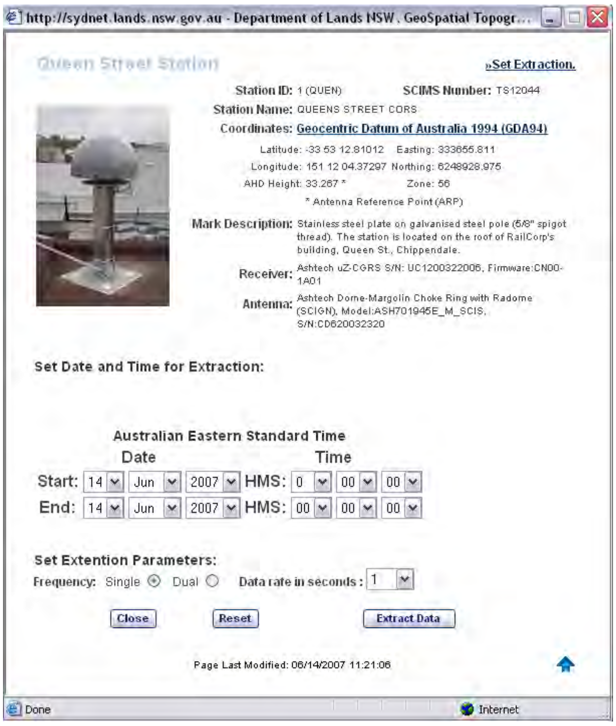
A distributed server can also serve as a local database server, or the two can be separately implemented. A distributed server parses data streams coming from reference stations and inserts the data into a database while the corresponding local database server monitors the quality of the data. The network administrator would want the quality check, the quality assurance, detection of cycle slips and outliers, monitoring high frequency variations in the double-differenced residuals, etc. A local database server sends such information to the main database server so that the network administrator can perform web-based monitoring by accessing the main database server only.

A client who wants post-processed positioning solutions can request the web server to obtain Receiver INdependent EXchange (RINEX) files. Then the web server retrieves RINEX files from a local database server via the main server. Figure 6 shows the webpage for extracting RINEX files from SydNET, which is a network of continuously operating reference stations located in Sydney, Australia (Department of Lands, 2006).

The main purpose of a local database server is to perform DB functions. A set of necessary DB functions is listed in Table 4.

Table 4. Local DB server functions

Function	Task
StoreDB/RetriveDB	To store and retrieve the parsed data into databases
MonitorDataQuality	To monitor the quality of data from reference stations
TransmitDataQuality	To transmit the monitored information to the main database server
RetrieveRINEX	To retrieve RINEX files from databases



http://sydnet.lands.nsw.gov.au - Department of Lands NSW . GeoSpatial Topogr...

Queen Street Station

»Set Extraction.

Station ID: 1 (QUEN) SCIMS Number: TS12044

Station Name: QUEENS STREET CORS

Coordinates: Geocentric Datum of Australia 1994 (GDA94)

Latitude: -33 53 12.81012 Easting: 333655.811

Longitude: 151 12 04.37297 Northing: 6248928.975

AHD Height: 33.267 * Zone: 56

* Antenna Reference Point (ARP)

Mark Description: Stainless steel plate on galvanised steel pole (5/8" spigot thread). The station is located on the roof of RailCorp's building, Queen St., Chippendale.

Receiver: Ashtech uZ-CGRS S/N: UC1200322006, Firmware:CN00-1A01

Antenna: Ashtech Dorne-Margolin Choke Ring with Radome (SCIGN), Model:ASH701945E_M_SCIS, S/N:CD620032320

Set Date and Time for Extraction:

Australian Eastern Standard Time

	Date				Time		
Start:	14	Jun	2007	HMS:	0	00	00
End:	14	Jun	2007	HMS:	00	00	00

Set Extention Parameters:

Frequency: Single ● Dual ○ Data rate in seconds : 1

Page Last Modified: 06/14/2007 11:21:06

Fig. 6 Web extraction for SydNET RINEX files

6. Main Server

The main server distributes computing resources to the distributed servers. Once a client requests server-based RTK solutions, the main server queries the main database server for the most suitable Delaunay triangle and allocates a distributed server to the client. The main server calculates the network-wide parameters, e.g. orbit errors and atmospheric parameters from the network-wide observations, empirical covariance parameters for Kriging, and so on. The main server determines whether precise IGS orbits or the network orbits must be used, unless there is a special request from the client. The set of main server functions is listed in Table 5.

Table 5. Main server functions

Function	Task
SelectRTK	To distribute computing resources
DownloadIGS	To download precise IGS orbits
InsertIGS	To insert IGS orbits into the main database server
NetworkOrbits	To compute orbit errors based on the network-wide observations
NetworkAtmosphere	To compute atmospheric errors based on the network-wide observations
NetworkParameters	To compute the network-wide parameters e.g. Kriging parameters

7. Main Database Server

In this proposed framework, the main database server requires Open Geospatial Consortium (OGC) Simple Feature Specification for Structured Query Language (SQL). OGC has published the specification in 1997, in order to propose a conceptual method for a SQL DBMS to deal with spatial data. Most DBMSs have implemented the spatial extensions recommended by OGC (Open Geospatial Consortium, 2006). For example, Oracle Spatial and Oracle Locator comply with the OGC specification. MySQL has implemented a spatial extension to follow the specification. PostgreSQL has a spatial module known as PostGIS. Therefore, unlike a traditional DBMS it is possible to store the geometry and topology of geographical features within a database.

Spatial operators and functions of a DBMS are powerful because spatial relationships can be retrieved rather than computed. This approach reduces the computational overhead significantly. For example, some spatial functions available include:

- *Area* returns the area of a polygon
- *Contains* indicates if Feature 1 completely contains Feature 2
- *Crosses* indicates if Feature 1 spatially crosses Feature 2
- *Disjoint* indicates if Feature 1 spatially disjoints from Feature 2 (i.e. does not intersect with each other)
- *Distance* returns the shortest distance between two points
- *Intersects* indicates if Feature 1 intersects Feature 2
- *Overlaps* indicates if Feature 1 overlaps Feature 2
- *Related* indicates if a given spatial relationship between Features 1 and 2 exists
- *Touches* indicates if Feature 1 touches Feature 2
- *Within* indicates if Feature 1 is within Feature 2

The main database server stores the coordinates, Voronoi polygons and Delaunay triangles of reference stations. Voronoi polygons are used to assign the nearest reference station for single-base RTK, while Delaunay triangles are used for network-RTK. The nearest reference station or the triangle that contains the rover is selected upon a client's request for server-side RTK. The main database server also stores the network-wide parameters: orbit errors, atmospheric parameters, and Kriging parameters. The set of main database server functions is listed in Table 6.

Table 6. Main DB server functions

Function	Task
StoreDataQuality/ RetrieveDataQuality	To store and retrieve the monitored information such as data quality checks, data quality assurance parameters, detected cycle slips and outliers, high frequency variations of double-differenced residuals, abnormal multipath effects, etc.
StoreIGS/ RetrieveIGS	To store and retrieve precise IGS orbits
StoreNetworkOrbits/ RetrieveNetworkOrbits	To store and retrieve the network-wide orbit errors
StoreNetworkAtmo/ RetrieveNetworkAtmo	To store and retrieve the network-wide atmospheric errors
StoreNetworkParam/ RetrieveNetworkParam	To store and retrieve the network-wide parameters, e.g. Kriging parameters
StoreFeature/ RetrieveFeature	To store and retrieve geographical features such as points, lines, and polygons
StoreReference/ RetrieveReference	To store and retrieve the geometry and topology of reference stations
StoreVoronoi/ RetrieveVoronoi	To store and retrieve the geometry and topology of Voronoi polygons
StoreDelaunay/ RetrieveDelaunay	To store and retrieve the geometry and topology of Delaunay triangles

8. Web Server

The web server is the portal for clients and the network administrator. Ordinary clients simply access the web server and submit their data streams via NTRIP in order to activate server-based RTK. Authorised clients request

RINEX files for their post-processing or request data streams from the server so that they can perform traditional RTK. JavaScript and eXtensible Markup Language (XML) play an important role in the web server because a web-based Application Programming Interface (API) for Asynchronous JavaScript and XML (AJAX) must be used by clients. Advanced clients or the network administrator can utilise the API for their web-programming, similar to programming with the Google Maps API (Lim, 2005). AJAX is not a new technology, but a new paradigm that uses JavaScript, XML, Document Object Model (DOM), and HTTP Request.

The objective of the proposed framework is that clients do not need to learn complicated GNSS algorithms or software. Precise point positioning and RTK services including "near real-time" applications are performed server-side and delivered to clients upon their HTTP Request. For example, near real-time structural deformation can be monitored on a client's website simply by sending an HTTP Request to the web server and receiving the output. The visualisation of the output can be interfaced by the web-based API. If a base map is necessary for the visualisation purpose, the client can query the web map server. This is currently feasible because of the AJAX technology. Furthermore, the surveying result, e.g. monitoring of the structural deformation, can be syndicated to anyone using the RSS technology. Likewise, the network administrator can develop tools for web-based monitoring of the CORS network.

9. Concluding Remarks

A new framework for server-based thin-client RTK services is proposed where distributed-computing is essential. The proposed framework is intended to extend the capability of the server to real-time data processing by integrating information and communication technologies and CORS network infrastructure. An optimal design that balances the computational overhead with communication latency is described.

The proposal provides end-users with "out-of-the-box" services, i.e. end-users obtain extensive GNSS capabilities and high productivity by overcoming the conventional constraints of an expensive set of GNSS receivers, proprietary data formats, user-installed carrier phase processing software, incomplete interoperability, limited communication links, and so on. The framework also utilises database and web technologies which enable servers to perform data management and spatial analysis, while end-users are able to syndicate data and create their own business models for data and result dissemination.

In order to demonstrate the usability of the framework, a prototype web-based data quality monitoring system has been developed on the Java platform. The monitoring system is able to display statistics of a local GNSS network and from network-RTK software in real time.

Acknowledgements

This paper is a revised version of the authors' paper presented at the Institute of Navigation Conference in 2007 (Lim & Rizos, 2007). The research presented in this paper was carried out with support from the Cooperative Research Centre for Spatial Information Project 1.4 "Integrating electricity, telecommunications and government infrastructure to deliver precise positioning services in regional areas", 2007-2010.

References

Dai, L. (2002), ***Augmentation of GPS with GLONASS and Pseudolite Signals for Carrier Phase-Based Kinematic Positioning***, PhD Thesis, School of Surveying & Spatial Information Systems, The University of New South Wales, Australia.

Department of Lands (2006), ***Online SydNET Data Access***, http://sydnet.lands.nsw.gov.au/, accessed 4 July 2007.

Fotopoulus, G. and Cannon, M.E. (2001), ***An Overview of Multi-Reference Stations Methods for Cm-Level Positioning***, GPS Solutions, 4(3): 1-10.

Kanzaki, M. (2006), ***Inverted RTK System and its Applications in Japan***, 12th IAIN Congress & 2006 Int. Symp. on GPS/GNSS, Jeju, Korea, 18-20 October, 455-458.

Leica (2007), ***Leica GPS Spidernet,*** http://www.leica-geosystems.com/corporate/en/products/gps_systems/lgs_4591.htm, accessed 16 August 2007.

Lim, S. (2005), ***Lecture Notes on Web 2.0 and AJAX***, School of Surveying & Spatial Information Systems, The University of New South Wales, Australia.

Lim, S. and Rizos, C. (2007). ***A New Framework for Server-Based and Thin-Client GNSS Operations for High Accuracy Applications in Surveying and Navigation***, U.S. ION GNSS, Fort Worth, Texas, 25-28 September, CD-ROM procs.

Open Geospatial Consortium (2006), ***OpenGIS Web Map Server Implementation Specification***, Ref. OGC 06-042.

Rizos, C., & Cranenbroeck, J.van. (2006), ***Alternatives to current GPS-RTK services***. 19th Int. Tech. Meeting of the Satellite Division of the U.S. Inst. of Navigation, Fort Worth, Texas, 26-29 September, 1219-1225.

Rizos, C., & Han, S. (2003), ***Reference station network based RTK systems - Concepts & progress***, Wuhan University Journal of Nature Sciences, 8(2B), 566-574.

Trimble (2007), ***Trimble Virtual Reference Systems***, http://www.trimble.com/vrs.shtml, accessed 16 August 2007.

QR Implementation of GNSS Centralized Approaches

A. Lannes
CNRS/LSS/[1] (France)

S. Gratton
CNES/OMP/[1] (France)

Abstract. When processing times series of global positioning data, one is led to introduce 'local variables,' which depend on the successive epochs of the time series, and a 'global variable' which remains the same all over these epochs with however possible state transitions from time to time. For example, the latter occur when some satellites appear or disappear. In the period defined by two successive transitions, the problem to be solved in the least-square sense is governed by a linear equation in which the key matrix has an angular block structure. This structure is well suited to recursive QR factorization. The corresponding techniques prove to be very efficient for GNSS data processing and quality control in real-time kinematics. The main objective of this paper is to show how the QR implementation of GNSS centralized approaches combines the advantages of all the methods developed hitherto in this field. The study is conducted by considering the simple case of continuous observations with a local-scale single baseline. The extension to networks is simply outlined.

Keywords. GNSS, DGPS, RTK. PPP. DIA. RAIM. LLL. Undifferential centralized data, reduced difference. Recursive Least Square (RLS). Quality control. Integer ambiguity resolution.

1 Introduction

In the traditional approach to differential GNSS, the satellite error terms are eliminated by forming the so-called single differences (SD). One then gets rid of the receiver error terms by computing, for each receiver to be considered, the corresponding double differences (DD): the discrepancies between the single differences (SD) and one of them taken as reference. Note that a similar situation arises in precise point positioning (PPP) with a single receiver. To handle the SD's in a homogeneous manner, one may equally well consider the discrepancies between the SD's and their mean value. By adopting the terminology introduced by Shi and Han (1992), one may then speak of 'centralized differences' (CD). At first sight, the ambiguities to be raised are then rational numbers (which are not necessarily integers). The GNSS community therefore considered that this idea could not be implemented easily. Fifteen years later, this principle was reintroduced in an independent manner (Lannes 2007a). In the corresponding approach, which referred to the same concept, but with another terminology, that of 'reduced difference' (RD), the difficulty related to rational ambiguities was overcome. The connection with the centralized undifferential method was then clarified (Lannes 2007b, 2008). In particular, it was shown that at any stage of the data assimilation procedure, it was possible to pass from the RD mode to the DD mode, and vice-versa. Shortly, the RD mode is well suited to quality control (see Sects. 6 in Lannes 2007b and 2008), while solving the rational-ambiguity problem amounts to solving a nearest-lattice-point problem of DD type (see Sect. 5.2 in Lannes 2007b).

When processing times series of global positioning data, one is led to introduce 'local variables' u_i which depend on the successive epochs t_i of the time series to be processed, and a 'global variable' v which remains the same all over these epochs with however possible state transitions from time to time. For example, the latter occur when some receiver-satellite signals appear or disappear. In the period defined by two successive transitions, the problem to be solved in the least-square (LS) sense is governed by a system of linear equations of the form

$$\left| \begin{array}{l} A_1 u_1 + B_1 v = b_1 \\ A_2 u_2 + B_2 v = b_2 \\ \quad\vdots \\ A_i u_i + B_i v = b_i \end{array} \right. \tag{1}$$

The definition of the variables u_i and v depends on the GNSS system under consideration. The components of u_i and v are real numbers, some components of v being integers (lying in $\mathbb{Z}$): the integer ambiguities of the problem.

[1] This work was also supported by the CERFACS (France): the European Centre for Research and Advanced Training in Scientific Computation.

In matrix terms, Eq. (1) can be displayed as follows:

$$\begin{bmatrix} A_1 & & & & B_1 \\ & A_2 & & & B_2 \\ & & \ddots & & \vdots \\ & & & A_i & B_i \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_i \\ v \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_i \end{bmatrix} \tag{2}$$

As specified in Sect. 6.3 of Björck 1996 (see also Golub and van Loan 1989, Bierman 1977), the angular block structure of matrix $[A\ B]$ is well suited to recursive QR factorization. When dealing with large-scale problems, numerical accuracy can thereby be improved.

More interestingly, the corresponding techniques prove to be very efficient for GNSS data processing and quality control; see, e.g., Tiberius (1998), Loehnert *et al* (2000), Chang and Guo (2005). As clarified in this paper, this is particularly the case for the GNSS centralized approaches, even when dealing with small-scale systems. In particular, in the quality-control procedures, the identification of biases is then made easier (see Sects. 4.3 and 3.3).

To introduce the reader to the QR implementation of these approaches, we now concentrate on the simple case of continuous observations in RTK mode with a local-scale single baseline (see, e.g., Table 1 in Feng and Li 2008). This problem can of course be dealt with as a special case of multiple-baseline networks with possibly missing data. In this paper, we will not proceed that way. Indeed, the corresponding theoretical framework would then mask the main guidelines of our contribution.

The RD approach presented in Lannes 2007ab addressed this particular GNSS system. The corresponding data-assimilation procedure was based on recursive least-square (RLS) filtering. In particular, the normal equation associated with Eq. (2) was solved with the aid of classical RLS techniques. The QR implementation of this procedure therefore remained to be done.

As revealed by the contents of the present paper, this implementation led us to clarify some important points. For example, the RD concept was revisited and generalized. The quality-control procedure was thereby strongly simplified. At last but not the least, the advantages of the RD and DD approaches were conjugated in a straightforward manner. As a result, the extension to general networks presented in Lannes 2008 is to be revisited accordingly. This will be done in a forthcoming paper.

1.1 Observational equations

The particular GNSS system examined in this paper is governed by the following observational equations (see, e.g., Sect. 14 in Strang and Borre 1997). For each frequency f_ν, for each receiver-satellite pair (r, s), and at each epoch t, the carrier-phase and code relations are respectively of the form

$$\begin{aligned} \phi_{\nu,t}(r,s) &= \rho_t(r,s) + c[\delta t_{\nu,t}(r) - \delta t_{\nu,t}(s)] \\ &\quad + \lambda_\nu[\varphi_{\nu,0}(r) - \varphi_{\nu,0}(s)] + \lambda_\nu N_\nu(r,s) + \varepsilon_{\nu,t}(r,s) \end{aligned} \tag{3}$$

$$p_{\nu,t}(r,s) = \rho_t(r,s) + c[\mathrm{d}t_{\nu,t}(r) - \mathrm{d}t_{\nu,t}(s)] + \epsilon_{\nu,t}(r,s) \tag{4}$$

In these equations, which are expressed in length units, $\rho_t(r,s)$ is the receiver-satellite range: the distance between satellite s (at the time $t - \tau$ where the signal is emitted) and receiver r (at the time t of its reception). The λ_ν's denote the wavelengths of the carrier waves; the integers $N_\nu(r,s)$ are the integer carrier-phase ambiguities. The instrumental delays and clock errors that for a given (ν, t) depend only on r and s are lumped together in the receiver and satellite error terms $\delta t_{\nu,t}(r)$, $\delta t_{\nu,t}(s)$ for the phase, and $\mathrm{d}t_{\nu,t}(r)$, $\mathrm{d}t_{\nu,t}(s)$ for the code (c is the speed of light); $\varphi_{\nu,0}(r)$ and $\varphi_{\nu,0}(s)$ are the initial phases (expressed in cycles) in receiver r and satellite s, respectively. The phase and code errors $\varepsilon_{\nu,t}(r,s)$ and $\epsilon_{\nu,t}(r,s)$ include both noise and residual model errors. Here, for clarity, the ionospheric and tropospheric delays are ignored (see Sect. 1.2 with a local-scale system).

For clarity, we now restrict ourselves to the single-frequency case. Equations (3) and (4) then reduce to

$$\begin{aligned} \phi_t(r,s) &= \rho_t(r,s) + c[\delta t_t(r) - \delta t_t(s)] \\ &\quad + \lambda[\varphi_0(r) - \varphi_0(s)] + \lambda N(r,s) + \varepsilon_t(r,s) \end{aligned} \tag{5}$$

$$p_t(r,s) = \rho_t(r,s) + c[\mathrm{d}t_t(r) - \mathrm{d}t_t(s)] + \epsilon_t(r,s) \tag{6}$$

It may be convenient to consider that a function $\vartheta(r,s)$, such as $\rho_t(r,s)$ for example, takes its values on a rectangular grid. When the system includes two receivers and n satellites (as it is the case here), this grid includes two lines and n columns; the values $\vartheta(r,s)$ then define a vector $\boldsymbol{\vartheta}$ of the 'observational space' $\mathbb{R}^{2n}$. These values are the components of $\boldsymbol{\vartheta}$ in the standard basis of $\mathbb{R}^{2n}$.

The variance-covariance matrix of the data vector $\boldsymbol{\psi} = \boldsymbol{\phi}$ (for the phase) or $\boldsymbol{\psi} = \boldsymbol{p}$ (for the code) is denoted by $V_{\boldsymbol{\psi}}$. Let $[\boldsymbol{\vartheta}]$ now be the column matrix whose entries are the components of $\boldsymbol{\vartheta}$. The size $\|\boldsymbol{\vartheta}\|_{\boldsymbol{\psi}}$ of a vector $\boldsymbol{\vartheta}$ of type $\boldsymbol{\psi}$ (for example, that of an observational residual of type $\boldsymbol{\psi}$) is defined via the relation

$$\|\boldsymbol{\vartheta}\|^2_{\boldsymbol{\psi}} := [\boldsymbol{\vartheta}]^{\mathrm{T}} V^{-1}_{\boldsymbol{\psi}} [\boldsymbol{\vartheta}] \tag{7}$$

1.2 SD equations

Let r_1 be the reference receiver, and r_2 be that of the user. Denote by $s_1, s_2, \ldots, s_n$ the satellites involved in the GNSS device at epoch t. A quantity such as

$$\vartheta^{(j)} := \vartheta(r_2\,, s_j) - \vartheta(r_1\,, s_j) \tag{8}$$

is then referred to as a single difference (SD) in ϑ. (In this paper, a notation such as $a := b$ means 'a is equal to b by definition.')

Adopting the notation defined in Eq. (8), we then obtain from Eq. (5) the SD phase equations

$$\phi_t^{(j)} = \rho_t^{(j)} + \lambda v^{(j)} + \alpha_t + \varepsilon_t^{(j)} \qquad (j = 1, \ldots, n) \tag{9}$$

where

$$v^{(j)} := N^{(j)} - N^{(1)} \tag{10}$$

and

$$\begin{aligned}\alpha_t := c[\delta t_t(r_2) - \delta t_t(r_1)] \\ + \lambda[\varphi_0(r_2) - \varphi_0(r_1)] + \lambda N^{(1)}\end{aligned} \tag{11}$$

According to its definition, α_t is an unknown receiver parameter shifted by an unknown number of wavelengths. The $n-1$ integers

$$v^{(2)}, v^{(3)}, \ldots, v^{(n)}$$

are the DD ambiguities of the problem; here, the latter are defined with regard to the first satellite of the list of visible satellites at the initialization epoch: $v^{(1)} = 0$. This pointed out, in the present approach, no 'usual double difference' is computed: the SD data are dealt with in a homogeneous manner (see Sect. 1.4).

The SD code equations are obtained from Eq. (6) in a similar manner:

$$p_t^{(j)} = \rho_t^{(j)} + a_t + \epsilon_t^{(j)} \qquad (j = 1, \ldots, n) \tag{12}$$

where

$$a_t := c[\mathrm{d}t_t(r_2) - \mathrm{d}t_t(r_1)] \tag{13}$$

1.3 Linearized SD equations

The position variable at epoch t, ξ_t, appears via the linearization of the single differences $\rho_t^{(j)}$ with respect to the position variable $\xi_{2;t}$ of the user receiver r_2. Here, we implicitly refer to the relation $\xi_{2;t} = \tilde{\xi}_{2;t} + \xi_t$. As

$$\rho_t^{(j)} = \rho_t(r_2, s_j) - \rho_t(r_1, s_j)$$

the linear expansion of $\rho_t^{(j)}$ is of the form

$$\rho_t^{(j)} = \tilde{\rho}_t^{(j)} + \left(\kappa_t^{(j)} \cdot \xi_t\right) \tag{14}$$

where $\kappa_t^{(j)}$ is the unitary vector that characterizes the direction $s_j \to r_2$ of the signal received at epoch t. The geometry-free SD equations (9) and (12) then yield the linearized SD equations

$$\left(\kappa_t^{(j)} \cdot \xi_t\right) + \lambda v^{(j)} + \alpha_t + \varepsilon_t^{(j)} = \tilde{\phi}_t^{(j)} \tag{15}$$

$$\left(\kappa_t^{(j)} \cdot \xi_t\right) + a_t + \epsilon_t^{(j)} = \tilde{p}_t^{(j)} \tag{16}$$

where (for $j = 1, \ldots, n$)

$$\tilde{\phi}_t^{(j)} := \phi_t^{(j)} - \tilde{\rho}_t^{(j)} \tag{17}$$

$$\tilde{p}_t^{(j)} := p_t^{(j)} - \tilde{\rho}_t^{(j)} \tag{18}$$

We now show how to express these equations in a more concise form. Denoting by $\{e_j\}_{j=1}^n$ the standard basis of $\mathbb{R}^n$, let us consider the vector

$$\vartheta := \sum_{j=1}^{n} \vartheta^{(j)} e_j \tag{19}$$

where the $\vartheta^{(j)}$'s are the SD's defined in Eq. (8); $\mathbb{R}^n$ is then regarded as the 'SD space.' Throughout this paper, to avoid any confusion, a function such as $\vartheta(r,s)$ is never denoted by the isolated symbol ϑ.

Let Γ_t be the operator defined by the relations

$$(\Gamma_t \xi_t)^{(j)} := \left(\kappa_t^{(j)} \cdot \xi_t\right) \qquad (j = 1, \ldots, n) \tag{20}$$

By construction, the elements of the j^{th} line of the matrix of Γ_t are the components of $\kappa_t^{(j)}$, i.e., the direction cosines of $\kappa_t^{(j)}$; this matrix includes n lines. Let us now denote by ζ be the vector of $\mathbb{R}^n$ whose components are all equal to unity. In terms of vectors, the linearized SD equations (15) and (16) can then be written as follows:

$$\Gamma_t \xi_t + \lambda v + \zeta \alpha_t + \varepsilon_t = \tilde{\phi}_t \tag{21}$$

$$\Gamma_t \xi_t + \zeta a_t + \epsilon_t = \tilde{p}_t \tag{22}$$

Note that ξ_t, α_t and a_t are local variables, whereas v is a global variable.

Let $[\vartheta]$ now be the column matrix whose entries are the components of ϑ. The size $\|\vartheta\|_\psi$ of a vector ϑ of type ψ (for example, that of an observational residual of type ψ) is defined via the relation

$$\|\vartheta\|_\psi^2 := [\vartheta]^{\mathrm{T}} V_\psi^{-1} [\vartheta] \tag{23}$$

where V_ψ is variance-covariance matrix of ψ:

$$V_\psi = \mathcal{S} V_{\boldsymbol{\psi}} \mathcal{S}^{\mathrm{T}} \tag{24}$$

Here, $\mathcal{S}$ is the matrix of the SD operator (see Eq. (8))

$$\mathcal{S}[\boldsymbol{\vartheta}] := [\vartheta] \tag{25}$$

Let us now introduce the Cholesky factorization

$$V_\psi^{-1} = U_\psi^{\mathrm{T}} U_\psi \tag{26}$$

where U_ψ is an invertible upper-triangular matrix. From Eq. (23), we then have

$$\|\vartheta\|_\psi^2 = [\vartheta]^{\mathrm{T}} U_\psi^{\mathrm{T}} U_\psi [\vartheta] = [U_\psi \vartheta]^{\mathrm{T}} [U_\psi \vartheta]$$

i.e.,

$$\|\vartheta\|_\psi^2 = [\vartheta_\psi]^{\mathrm{T}} [\vartheta_\psi] \tag{27}$$

where

$$[\vartheta_\psi] := U_\psi [\vartheta] \tag{28}$$

According to these equations, the size of a vector ϑ of type ψ is equal to the size of ϑ_ψ in $\mathbb{R}^n$:

$$\|\vartheta\|_\psi^2 = \|\vartheta_\psi\|^2 \tag{29}$$

As clarified in Sect. 1.4, this trick proves to play a key role in the approach presented in this paper.

1.4 Statement of the problem

Let t_1 be the initialization epoch of the 'current run' $[t_1, \ldots, t_i]$. According to Eqs. (21) and (22), the problem is to minimize the objective functional

$$\begin{aligned} f(\xi_1, \ldots, \xi_i; v; \alpha_1, \ldots, \alpha_i; a_1, \ldots, a_i) & \\ := \textstyle\sum_{\iota=1}^{i} \|\tilde{\phi}_\iota - (\Gamma_\iota \xi_\iota + \lambda v) - \zeta \alpha_\iota\|^2_{\phi_\iota} & \qquad (30) \\ + \|\tilde{p}_\iota - \Gamma_\iota \xi_\iota - \zeta a_\iota\|^2_{p_\iota} & \end{aligned}$$

where $\xi_\iota \equiv \xi_{t_\iota}$, and likewise for α_ι, a_ι, $\tilde{\phi}_\iota$, $\tilde{p}_\iota$ and Γ_ι. In our approach, this is done in two steps. The first step is to minimize f in α_ι and a_ι for $\iota = 1, \ldots, i$. As clarified below, this operation corresponds to the notion of 'reduction.'

1.4.1 Reduced equations

Let us first concentrate on the phase terms. For clarity, let us then set $\vartheta := \tilde{\phi}_\iota - (\Gamma_\iota \xi_\iota + \lambda v)$. The optimal estimate of α_ι is then the real number $\alpha_\circ$ for which the minimum of $\|\vartheta - \zeta\alpha\|_\phi$ in α is attained. From Eq. (29), we have

$$\|\vartheta - \zeta\alpha\|^2_\phi = \|\vartheta_\phi - \zeta_\phi \alpha\|^2$$

where ϕ stands for ϕ_ι. As a result, $\alpha_\circ$ is the solution of the normal equation

$$[\zeta_\phi]^{\mathrm{T}}[\zeta_\phi]\alpha = [\zeta_\phi]^{\mathrm{T}}[\vartheta_\phi]$$

i.e.,

$$\alpha_\circ = \frac{[\zeta_\phi]^{\mathrm{T}}[\vartheta_\phi]}{[\zeta_\phi]^{\mathrm{T}}[\zeta_\phi]}$$

It follows that

$$\vartheta_\phi - \zeta_\phi \alpha_\circ = \mathcal{R}_\phi \vartheta$$

where (here, for $\psi = \phi \equiv \phi_\iota$)

$$\mathcal{R}_\psi \vartheta := \vartheta_\psi - \frac{[\zeta_\psi]^{\mathrm{T}}[\vartheta_\psi]}{[\zeta_\psi]^{\mathrm{T}}[\zeta_\psi]}\, \zeta_\psi \qquad (31)$$

Consequently (see Eq. (30)):

$$\min_{\alpha_\iota \in \mathbb{R}} \|\tilde{\phi}_\iota - (\Gamma_\iota \xi_\iota + \lambda v) - \zeta \alpha_\iota\|^2_{\phi_\iota} = \|\mathcal{R}_{\phi_\iota}[\tilde{\phi}_\iota - (\Gamma_\iota \xi_\iota + \lambda v)]\|^2$$

Likewise, for the code terms,

$$\min_{a_\iota \in \mathbb{R}} \|\tilde{p}_\iota - \Gamma_\iota \xi_\iota - \zeta a_\iota\|^2_{p_\iota} = \|\mathcal{R}_{p_\iota}(\tilde{p}_\iota - \Gamma_\iota \xi_\iota)\|^2$$

We are thus led to minimize the 'reduced functional'

$$\begin{aligned} f_{\mathrm{r}}(\xi_1, \ldots, \xi_i; v) & \\ := \textstyle\sum_{\iota=1}^{i} \|\mathcal{R}_{\phi_\iota}[\tilde{\phi}_\iota - (\Gamma_\iota \xi_\iota + \lambda v)]\|^2 & \qquad (32) \\ + \|\mathcal{R}_{p_\iota}(\Gamma_\iota \xi_\iota - \tilde{p}_\iota)\|^2 & \end{aligned}$$

The 'reduced equations' to be solved in the usual LS sense can therefore be displayed as follows:

$$\mathcal{R}_{\phi_\iota}(\Gamma_\iota \xi_\iota + \lambda v) = \mathcal{R}_{\phi_\iota} \tilde{\phi}_\iota \qquad (33)$$

$$\mathcal{R}_{p_\iota} \Gamma_\iota \xi_\iota = \mathcal{R}_{p_\iota} \tilde{p}_\iota \qquad (34)$$

1.4.2 Reduction operator

Let us concentrate on the 'reduction operator' (31). For clarity, let us set

$$\vartheta_{\mathrm{r};\psi} := \mathcal{R}_\psi \vartheta \qquad (35)$$

To give a more concrete idea of the action of this operator, let us now consider the typical situation where the variance-covariance matrix of the observational data of type $\boldsymbol{\psi}$ is of the form (see Liu 2002)

$$V_{\boldsymbol{\psi}} = \mathrm{diag}\big(\eta(r, s)\, \sigma^2_{\boldsymbol{\psi}}\big) \qquad (36)$$

Here, $\sigma^2_{\boldsymbol{\psi}}$ is a 'reference variance;' $\eta(r, s)$ is a nonnegative weight function. The variance-covariance matrix of the SD data is then given by the relation (see Eq. (24))

$$V_\psi = \mathrm{diag}(\eta_j \sigma^2_{\boldsymbol{\psi}}) \qquad \eta_j := \eta(r_1\,, s_j) + \eta(r_2\,, s_j) \qquad (37)$$

From Eq. (26), we then have

$$U_\psi = \mathrm{diag}\left(\frac{1}{\sqrt{\eta_j}\, \sigma_{\boldsymbol{\psi}}}\right) \qquad (38)$$

hence, from Eq. (28),

$$\vartheta^{(j)}_\psi = \frac{1}{\sqrt{\eta_j}\, \sigma_{\boldsymbol{\psi}}}\, \vartheta^{(j)} \qquad \zeta^{(j)}_\psi = \frac{1}{\sqrt{\eta_j}\, \sigma_{\boldsymbol{\psi}}}\, \zeta^{(j)}$$

As $\zeta^{(j)} = 1$ for all j, we then have

$$[\zeta_\psi]^{\mathrm{T}}[\vartheta_\psi] = \frac{1}{\sigma^2_{\boldsymbol{\psi}}} \sum_{j=1}^{n} \frac{1}{\eta_j} \vartheta^{(j)} \qquad [\zeta_\psi]^{\mathrm{T}}[\zeta_\psi] = \frac{1}{\sigma^2_{\boldsymbol{\psi}}} \sum_{j=1}^{n} \frac{1}{\eta_j}$$

It then follows from Eqs. (35) and (31) that the components of $\vartheta_{\mathrm{r};\psi}$ are given by the formula

$$\vartheta^{(j)}_{\mathrm{r};\psi} = \frac{\vartheta^{(j)} - \vartheta^{(0)}}{\sigma_{\psi j}} \qquad \sigma_{\psi j} := \sqrt{\eta_j}\, \sigma_{\boldsymbol{\psi}} \qquad (39)$$

where

$$\vartheta^{(0)} := \sum_{j=1}^{n} \mu_j \vartheta^{(j)} \qquad \mu_j := \frac{\frac{1}{\eta_j}}{\sum_{k=1}^{n} \frac{1}{\eta_k}} \qquad (40)$$

Note that $\sigma_{\psi j}$ is the standard deviation of the single-difference $\psi^{(j)}$. With regard to the SD weights $1/\eta_j$ or $1/\sigma^2_{\psi j}$, $\vartheta^{(0)}$ is a 'barycentric single difference:'

$$\sum_{j=1}^{n} \frac{\vartheta^{(j)} - \vartheta^{(0)}}{\sigma^2_{\psi j}} = 0$$

According to its notation, this virtual single difference is associated with a virtual satellite s_0. The n 'virtual double differences' $\vartheta^{(j)} - \vartheta^{(0)}$ can thus be regarded as the 'centralized values' of the $\vartheta^{(j)}$'s (Shi and Han 1992), or equally well, as the 'reduced values' of the $\vartheta^{(j)}$'s (Lannes 2007ab). Indeed, the minimum of

$$\sum_{j=1}^{n} \frac{(\vartheta^{(j)} - \omega)^2}{\sigma^2_{\psi j}} \qquad (\omega \in \mathbb{R})$$

is obtained for $\omega = \vartheta^{(0)}$. In other terms, in a concrete manner, the action of $\mathcal{R}_\psi$ consists in performing this type of reduction.

1.5 Contents

As specified in Sect. 2, the reduced equations (33) and (34) lead to a linear system of type (2). The block matrices A_i, B_i and b_i are then defined, and likewise for the local variables $u_1, u_2, \ldots, u_i$ and the global variable v. The components of v are then the float ambiguities of the problem.

The float solution $\hat{v}$ is refined recursively, epoch-by-epoch, with the aid of the QR method. This method is introduced in Sect. 3.1, and fully described in Sect. 3.2. The selected QR implementation is based on 'Givens rotations' (see, e.g., Björck 1996); the corresponding operations can thus be stored in memory very easily. This is very useful for the variational method presented in Sect. 3.3. As the latter is basically involved in the quality-control procedures (see Sect. 4), the efficiency of the DIA method presented in Lannes 2007b is thereby improved. The state transitions induced by the appearance and/or the disappearance of some satellites are examined in Sects. 3.4 and 3.5, respectively. As specified in Sect. 3.6, the inverse of the variance-covariance matrix of $\hat{v}$ is directly provided by the QR method. The procedure that yields the integer-ambiguity solution $\check{v}$ is described in that section.

This study is illustrated with dual-frequency examples (Sect. 5). Some comments on the key points of our contribution, and its extension to GNSS networks are to be found in Sect. 6.

2 Block matrices of the global RD equation

The reduced equations (33) and (34) lead to an equation of type (2). We now clarify this point explicitly. The extension to the dual-frequency case is straightforward (see Sect. 5).

The local variable u_i then reduces to the position variable ξ_i. The block matrix A_i is then defined as follows:

$$A_i = \begin{bmatrix} \mathcal{R}_{\phi_i}\Gamma_i \\ \mathcal{R}_{p_i}\Gamma_i \end{bmatrix} \tag{41}$$

Note that $\mathcal{R}_\psi\Gamma_i$ is obtained by applying the reduction operator $\mathcal{R}_\psi$ to each column vector of Γ_i (see Eq. (31) and Sect. 1.4.2). The corresponding data block of Eq. (2) is then

$$b_i = \begin{bmatrix} \mathcal{R}_{\phi_i}\tilde{\phi}_i \\ \mathcal{R}_{p_i}\tilde{p}_i \end{bmatrix} \tag{42}$$

Let $\bar{\mathrm{S}}_i := \{s_1, s_2, \ldots, s_{\bar{n}_i}\}$ be the series of satellites involved in the observational process until epoch t_i included. A given satellite may disappear and reappear in the same run. Such a satellite is then regarded as a new satellite. In other words, whenever this occurs, a new satellite is added at the end of this series. The n_i satellites of epoch t_i form a subset S_i of $\bar{\mathrm{S}}_i$: $n_i \leq \bar{n}_i$.

To introduce the reader to what is essential, we first restrict ourselves to the case where no satellite appears or disappears in the current run $[t_1, \ldots, t_i]$: no state transition in this interval. The entries of the global variable v are then the ambiguities $v^{(2)}, v^{(3)}, \ldots, v^{(n_i)}$ with $n_i = \bar{n}_i$ (see Eq. (10)). As clarified in Sect. 3.4, it is recommended to class these ambiguities in reverse order. For example, for $n_i = 7$, the global variable v is then explicitly defined as the column matrix (with 6 entries)

$$v = \begin{bmatrix} v^{(7)} \\ v^{(6)} \\ \vdots \\ v^{(3)} \\ v^{(2)} \end{bmatrix} \tag{43}$$

The phase block of B_i is then of the form (see Eq. (33)):

$$\left[B_i\right]_{\phi_i} = \mathcal{R}^{[n_i]}_{\phi_i} \begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \lambda \\ \cdot & \cdot & \cdot & \cdot & \lambda & \cdot \\ \cdot & \cdot & \cdot & \lambda & \cdot & \cdot \\ \cdot & \cdot & \lambda & \cdot & \cdot & \cdot \\ \cdot & \lambda & \cdot & \cdot & \cdot & \cdot \\ \lambda & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \qquad (n_i = 7) \tag{44}$$

Here, the dots stand for 0. This matrix includes n_i lines (corresponding to the n_i visible satellites of the system), and $n_i - 1$ columns (corresponding to the $n_i - 1$ ambiguities of the problem). The notation $\mathcal{R}^{[n_i]}_{\phi_i}$ means that the reduction operation is performed on vectors of $\mathbb{R}^{n_i}$. Here, as the reference satellite s_1 of the current run is visible, the first line is nought (see Eqs. (9) and (10)).

Note that the code block of B_i is nought: $\left[B_i\right]_{p_i} = 0$.

3 QR method

We first introduce the reader to the notion of QR factorization (Sect. 3.1). We then show how to solve Eq. (2) in a recursive manner (Sect. 3.2). The corresponding variational aspects are presented in Sect. 3.3. We then specify how to handle the ambiguities when some satellites appear and/or disappear (Sects. 3.4 and 3.5, respectively). Finally, Sect. 3.6 is devoted to the QR aspects concerning the integer ambiguity problem.

3.1 QR factorization

Let us consider the following general LS problem: minimize, with the Euclidean norm,

$$\|\mathrm{A}x - y\|^2_{\mathbb{R}^\mathrm{m}} \quad (\mathrm{A} \in \mathbb{R}^{\mathrm{m}\times\mathrm{n}},\ \mathrm{m} \geq \mathrm{n},\ \mathrm{rank}\ \mathrm{A} = \mathrm{n})$$

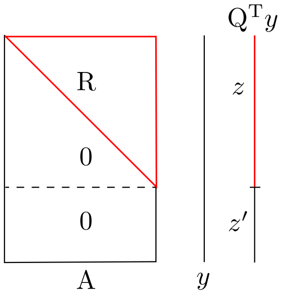


Fig. 1 *LS solution via QR factorization.* The action of $\mathrm{Q^T}$ on A and y yields the basic QR structure sketched here: the upper-triangular matrix R and the column matrix z. The solution of the equation $\mathrm{A}x = y$ in the LS sense is then given by the formula $x = \mathrm{R}^{-1}z$ (see Eq. (46)).

With regard to numerical accuracy, the best way to solve this problem is to use a method based on the QR factorization of A (see, e.g., Björck 1996):

$$\mathrm{A} = \mathrm{Q}\begin{bmatrix} \mathrm{R} \\ 0 \end{bmatrix} \tag{45}$$

where $\mathrm{R} \in \mathbb{R}^{\mathrm{n}\times\mathrm{n}}$ is an upper triangular matrix with positive diagonal terms, and $\mathrm{Q} \in \mathbb{R}^{\mathrm{m}\times\mathrm{m}}$ is an orthogonal matrix: $\mathrm{Q^TQ} = \mathrm{I_m}$ (the identity matrix on $\mathbb{R}^{\mathrm{m}}$). We thus have

$$\begin{aligned} \|\mathrm{A}x - y\|^2_{\mathbb{R}^{\mathrm{m}}} &= \|\mathrm{Q^T}(\mathrm{A}x-y)\|^2_{\mathbb{R}^{\mathrm{m}}} \\ &= \left\|\mathrm{Q^TQ}\begin{bmatrix} \mathrm{R} \\ 0 \end{bmatrix} x - \mathrm{Q^T}y\right\|^2_{\mathbb{R}^{\mathrm{m}}} \end{aligned}$$

Setting $\mathrm{Q^T}y = z + z'$ where $z \in \mathbb{R}^{\mathrm{n}}$ (see Fig. 1), it follows that

$$\|\mathrm{A}x - y\|^2_{\mathbb{R}^{\mathrm{m}}} = \|\mathrm{R}x - z\|^2_{\mathbb{R}^{\mathrm{n}}} + \|z'\|^2_{\mathbb{R}^{\mathrm{m-n}}} \tag{46}$$

The LS solution is therefore given by the relation

$$\hat{x} = \mathrm{R}^{-1}z \tag{47}$$

The problem can thereby be solved by back substitution. In the case where x is confined to $\mathbb{Z}^{\mathrm{n}}$, the solution of the problem is therefore defined as follows:

$$\check{x} = \operatorname*{argmin}_{x\in\mathbb{Z}^{\mathrm{n}}} \|\mathrm{R}(x-\hat{x})\|^2_{\mathbb{R}^{\mathrm{n}}} \tag{48}$$

Indeed, $\mathrm{R}x - z = \mathrm{R}(x - \hat{x})$.

According to Eq. (45), QR factorization consists in finding an operator $\mathrm{Q^T}$ (and thereby an operator Q) such that $\mathrm{Q^TA}$ has the block structure $[\mathrm{R}\ \ 0]^{\mathrm{T}}$ sketched in Fig. 1. This operator is defined as a product of elementary orthogonal transformations. In the implementation presented in this paper, the latter are Givens rotations (see Eqs. (2.3.10) to (2.3.13) in Björck 1996). Premultiplication of A and y by such a rotation matrix affects only rows k and ℓ of A and d. This matrix is defined so that, for $(\mathrm{a}_k^2 + \mathrm{a}_\ell^2) \neq 0$,

$$\begin{bmatrix} \mathrm{c} & \mathrm{s} \\ -\mathrm{s} & \mathrm{c} \end{bmatrix}\begin{bmatrix} \mathrm{a}_k \\ \mathrm{a}_\ell \end{bmatrix} = \begin{bmatrix} \mathrm{a} \\ 0 \end{bmatrix} \tag{49}$$

where

$$\mathrm{a} = (\mathrm{a}_k^2 + \mathrm{a}_\ell^2)^{1/2} \tag{50}$$

It is easy to check that the cosine and sinus values c and s are then given by the following formulas

$$\mathrm{c} = \mathrm{a}_k/\mathrm{a} \qquad \mathrm{s} = \mathrm{a}_\ell/\mathrm{a} \tag{51}$$

Note that m−1 Givens rotations are required for the first column of A, m−2 for the second, and so on (see Fig. 1). It is important to point out that that the action of $\mathrm{Q^T}$ can be stored in memory as the sequence of the successive (cosine, sinus) pairs (c, s) characterizing the successive Givens rotations involved in this operation.

3.2 Recursive QR factorization

We now show how to solve, in the LS sense and recursively, the equation (2) induced by the reduced equations (33) and (34).

Let us first consider the initialization epoch: epoch 1. The problem is then solved in two steps (see Fig. 2). The Givens rotations of the first step are those required for finding the upper triangular matrix K_1. The modified version of B_1 thus obtained includes an upper block L_1 and a lower block L'_1. Likewise, the modified version of b_1 includes two column submatrices: c_1 and c'_1. The Givens rotations of the second step yield the upper triangular matrix R_1; c'_1 then yields $(d_1\,, d'_1)$; see Fig. 2. Note that K_1, L_1 and c_1 are not affected by these rotations. The global solution is then obtained by back substitution via the formula $\hat{v} = R_1^{-1}d_1$. The local solution can then be also computed by back substitution: $\hat{u}_1 = K_1^{-1}(c_1 - L_1\hat{v})$.

The first step of the next epoch (epoch 2) is similar to that of epoch 1: one thus obtains the upper triangular matrix K_2. The modified version of B_2 then includes an upper block L_2 and a lower block L'_2. Likewise, the modified version of b_2 includes two column submatrices: c_2 and c'_2 (see Fig. 2). The Givens rotations of the second step then operate on $(R_1\,, L'_2)$ and $(d_1\,, c'_2)$ so as to transform L'_2 into a zero block matrix. One thus gets R_2 and $(d_2\,, d'_2)$; $\hat{v}$ is then updated via the relation $\hat{v} = R_2^{-1}d_2$. The local solution at epoch 2 can then be computed: $\hat{u}_2 = K_2^{-1}(c_2 - L_2\hat{v})$.

In summary, one thus operates, recursively, with the key structure shown in Fig. 3: K_i, $(L_i\,, L'_i)$ and $(c_i\,, c'_i)$ are computed from A_i, B_i and b_i, R_i and $(d_i\,, d'_i)$ being then computed from $(R_{i-1}\,, L'_i)$ and $(d_{i-1}\,, c'_i)$. We then have

$$\begin{bmatrix} K_i & L_i \\ \cdot & R_i \end{bmatrix}\begin{bmatrix} \hat{u}_i \\ \hat{v} \end{bmatrix} = \begin{bmatrix} c_i \\ d_i \end{bmatrix} \tag{52}$$

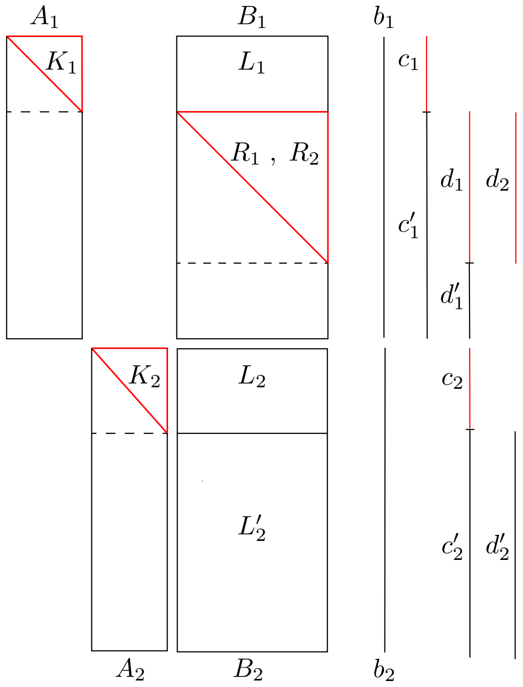


Fig. 2 *LS solution via recursive QR factorization.* The principle of the recursive QR method is sketched here for the first two epochs: epoch 1 with the input block matrices A_1 , B_1 and the data column matrix b_1; epoch 2 with the input block matrices A_2 , B_2 and the data column matrix b_2. The initialization process is performed in two steps: K_1 , $(L_1\,, L'_1)$, $(c_1\,, c'_1)$ are built in the first step (see text for L'_1), whereas R_1 , $(d_1\,, d'_1)$ are built in the second. The global float solution is then found by back substitution: $\hat{v} = R_1^{-1} d_1$. The local solution is then given by the formula $\hat{u}_1 = K_1^{-1}(c_1 - L_1\hat{v})$. Likewise, at the next epoch, one first builds K_2 , $(L_2\,, L'_2)$, $(c_2\,, c'_2)$, and then R_2, $(d_2\,, d'_2)$; $\hat{v}$ is then updated via the relation $\hat{v} = R_2^{-1} d_2$. The local solution at epoch 2 can then be computed: $\hat{u}_2 = K_2^{-1}(c_2 - L_2\hat{v})$.

hence $\hat{v} = R_i^{-1} d_i$ and $\hat{u}_i = K_i^{-1}(c_i - L_i\hat{v})$. The detailed implementation of this process must of course take account of the fact the code block of B_i is nought.

3.3 Variational calculation

We now answer to the following question: what are the variations $\Delta\hat{u}_i$ and $\Delta\hat{v}$ induced by a variation Δb_i of b_i (at epoch t_i)? From Eq. (2), these variations are the u-v components at epoch t_i of the LS solution of the equation

$$\begin{bmatrix} A_1 & & & & B_1 \\ & A_2 & & & B_2 \\ & & \ddots & & \vdots \\ & & & A_i & B_i \end{bmatrix} \begin{bmatrix} \Delta u_1 \\ \Delta u_2 \\ \vdots \\ \Delta u_i \\ \Delta v \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \Delta b_i \end{bmatrix}$$

By construction, the quantities $\Delta d_1, \ldots, \Delta d_{i-1}$ induced by this equation are nought. The problem is therefore the same as previously, Δd_i being then computed from $\Delta c'_i$ with $\Delta d_{i-1} = 0$. This is why it is recommended to store in memory the sequence of the successive pairs (c, s) characterizing the Givens operators involved in the two QR steps of epoch t_i (see Fig. 2 and Eqs. (51) & (50)).

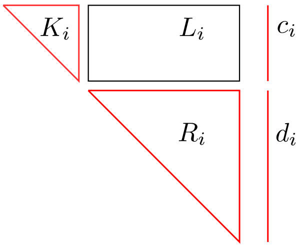


Fig. 3 *Recursive QR triangular structure.* According to the principle of the recursive QR method sketched in Fig. 2, the calculation of R_i and d_i requires to have kept in memory the upper triangular matrix R_{i-1} and the column matrix d_{i-1} (see text).

3.4 Handling the ambiguities when some satellites appear

As shown in Eq. (43), the ambiguities are put in reverse order. When some satellites appear at epoch t_i, the first columns of B_i can then be processed as the last columns of A_i (see Fig. 2). To get R_i and d_i, one then proceeds as illustrated in Fig. 4.

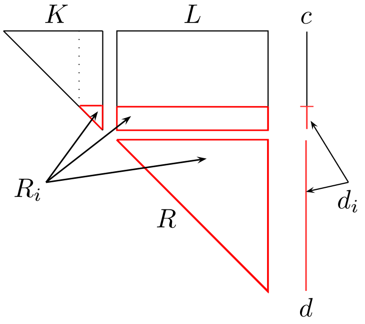


Fig. 4 *Handling additional ambiguities.* When satellites appear at epoch t_i, the first columns of B_i are processed as the last columns of A_i. The recursive QR operation then yields the quantities K, L, c, R and d. To get R_i and d_i, one then proceeds as illustrated here.

3.5 Handling the ambiguities when some satellites disappear

Let us first consider the case where the reference satellite of the current run disappears at epoch t_i. For example, with regard to the situation corresponding to Eq. (44), the phase block of B_i then becomes

$$[B_i]_\phi = \mathcal{R}_\phi^{[n_i]} \begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \lambda \\ \cdot & \cdot & \cdot & \cdot & \lambda & \cdot \\ \cdot & \cdot & \cdot & \lambda & \cdot & \cdot \\ \cdot & \cdot & \lambda & \cdot & \cdot & \cdot \\ \cdot & \lambda & \cdot & \cdot & \cdot & \cdot \\ \lambda & \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix} \qquad (n_i = 6) \qquad (53)$$

The calculation of R_i and d_i is then performed as usually. Indeed, as the ambiguities to be considered remain the same, R_{i-1} and d_{i-1} must not be modified.

Let us now consider the case where, for example, the satellites s_7 and s_6 disappear at epoch t_i. The ambiguities v_7 and v_6 of Eq. (43) can then be removed. The phase block of B_i is then of the form (see Eq. (44))

$$[B_i]_\phi = \mathcal{R}_\phi^{[n_i]} \begin{bmatrix} \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \lambda \\ \cdot & \cdot & \lambda & \cdot \\ \cdot & \lambda & \cdot & \cdot \\ \lambda & \cdot & \cdot & \cdot \end{bmatrix} \qquad (n_i = 5) \qquad (54)$$

In the calculation of the upper triangular matrix R_i, R_{i-1} is then simply updated by removing its first two lines and first two columns. Likewise, in the calculation of d_i, the first two entries of d_{i-1} are then to be removed.

Let us now consider the case where, for example, satellites s_5 and s_3 disappear at epoch t_i, the phase block of B_i is then of the same as that defined in Eq. (54); R_{i-1} and d_{i-1} must then be modified as specified below.

One first performs the permutation

$$\begin{bmatrix} v^{(7)} \\ v^{(6)} \\ v^{(5)} \\ v^{(4)} \\ v^{(3)} \\ v^{(2)} \end{bmatrix} \longrightarrow \begin{bmatrix} v^{(5)} \\ v^{(3)} \\ v^{(7)} \\ v^{(6)} \\ v^{(4)} \\ v^{(2)} \end{bmatrix} \qquad (55)$$

The columns of R_{i-1} are then permuted accordingly. As the matrix thus obtained, R'_{i-1}, is no longer upper triangular, one then performs Givens rotations on R'_{i-1} and d_{i-1} so that R'_{i-1} becomes upper triangular: $R'_{i-1} \to R''_{i-1}$, $d_{i-1} \to d''_{i-1}$. To complete the process, one then removes the first two lines and first two columns of R''_{i-1}, as well as the first two entries of d''_{i-1}.

3.6 Integer-ambiguity resolution

Let $\hat{v}$ be the float solution at epoch t_i, and $\mathfrak{n}$ be the number of its components. In single-frequency mode, depending on whether the reference satellite of the run $[t_1, t_i]$ is visible or not, $\mathfrak{n}$ is equal to $n_i - 1$ or n_i (respectively). The ambiguity solution is then defined by the relation (see Eq. (48))

$$\check{v} = \underset{v \in \mathbb{Z}^{\mathfrak{n}}}{\operatorname{argmin}} \|R_i(v - \hat{v})\|^2_{\mathbb{R}^{\mathfrak{n}}} \qquad (56)$$

According to this formula, $\check{v}$ is the point of $\mathbb{Z}^{\mathfrak{n}}$ closest to $\hat{v}$, the distance being that induced by the quadratic form

$$q(v) := \|R_i v\|^2_{\mathbb{R}^{\mathfrak{n}}} = v^{\mathrm{T}}[R_i^{\mathrm{T}} R_i] v \qquad (57)$$

Note that $R_i^{\mathrm{T}} R_i$ is the inverse of the variance-covariance matrix of $\hat{v}$:

$$R_i^{\mathrm{T}} R_i = V_{\hat{v}}^{-1} \qquad (58)$$

The QR method thus provides the Cholesky factor R_i of the matrix of q directly. This is not the case in the usual RLS filtering techniques. Indeed, the latter provide $V_{\hat{v}}$ which is then to be inverted.

The nearest-lattice-point problem (56) is solved in two steps (see, e.g., Agrell et al. 2002). One first searches a 'reduced basis' of $\mathbb{Z}^{\mathfrak{n}}$ in which the matrix of q is as diagonal as possible. The problem is then solved in this basis by using the corresponding 'reduced form' of R_i: $\bar{R}_i$; the integer-valued solution $\check{v}$ is then expressed in the original basis.

The first step corresponds to a decorrelation process. The decorrelation methods to be implemented must somehow refer to the principles of the LLL algorithm (an algorithm devised by Lenstra, Lenstra and Lovàsz in 1982). Here, as the QR recursive process provides R_i directly, the LLL implementations of Luk and Tracy (2008) are well suited to the problem. Denoting by $\bar{r}_{k,\ell}$'s the matrix elements of $\bar{R}_i$, the following conditions can thus be imposed:

(i) $\quad \bar{r}_{k,k} > 2|\bar{r}_{k,\ell}| \qquad$ (for $1 \leq k < \ell \leq \mathfrak{n}$)

(ii) $\quad \bar{r}^2_{k,k} \geq (\omega - 1/4)\bar{r}^2_{k-1,k-1} \qquad$ (for $2 \leq k \leq \mathfrak{n}$)

with $1/4 < \omega < 1$. In practice, to speed up the second-step procedure, ω is set equal to 0.999. Note that Condition (ii) is not necessarily imposed in other decorrelation methods (see, e.g., Xu 2001).

When in the data assimilation process, $\check{v}$ becomes consistent with the model, the ambiguities are said to be fixed. The local variable $\hat{u}_i$ is then refined via a fixed least-squares (FLS) process, i.e., a process in which the ambiguities are fixed at these values. Again, the QR method is well suited to solving these problems.

4 Quality control

To prevent that biases on the SD data propagate undetected into the ambiguity solution and the positioning results, particular methods have been developed. The biases are first 'detected,' then 'identified,' and finally the results are 'adapted' consequently (e.g., Teunissen 1990, Hewitson et al. 2004). Note that these DIA methods are to be implemented in all the modes to be considered: LS, RLS and FLS.

The DIA method presented in this section is a simplified version of that presented in Lannes 2007b. Its identification principle is 'local,' in the sense that the biases thus identified concern only the data of the current epoch. In the present version, the corresponding analysis is based on the results provided by the QR process at that epoch. When the ambiguities are not fixed, the adaptation principle is global: the local position, the current biases, the current float ambiguities and the current QR triangular structure (sketched in Fig. 3) are updated in the global frame of the QR recursive process, without any approximation. This was not completely the case in Lannes 2007b.

4.1 Local identification

The identification principle is based on the analysis of the residual at epoch t_i:

$$w_i := b_i - (A_i \hat{u}_i + B_i \hat{v}) \qquad (59)$$

Note that $\hat{u}_i$ and $\hat{v}$ depend on b_i in a linear manner. Let us now denote by y_i the column matrix of the SD data corrected from the terms due to linearization (see Eqs. (42))

$$y_i := \begin{bmatrix} \tilde{\phi}_i \\ \tilde{p}_i \end{bmatrix} \qquad (60)$$

In what follows, H_i is the operator that yields w_i from y_i (see Eqs. (42) and (59)):

$$w_i = H_i y_i \qquad (61)$$

For clarity, we now omit subscript i. Denoting by w_p and w_ϕ the code and phase components of w (respectively), we then have, in single-frequency mode,

$$\|w\|^2 := \|w_\phi\|^2 + \|w_p\|^2 \qquad (62)$$

where $\|w_\psi\|^2 = \sum_{j_\psi=1}^{n} |w_{j_\psi}|^2$ for $\psi = p$ or ϕ. When $\|w\|^2$ is too large (see Sect. 4.3), we then search to identify, in the SD data y, a global bias of the form

$$z = \begin{bmatrix} \sum_{j_\phi \in \Omega_\phi} \beta_{j_\phi} e_{j_\phi} \\ \sum_{j_p \in \Omega_p} \beta_{j_p} e_{j_p} \end{bmatrix} \qquad (63)$$

The 'outlier sets' Ω_ϕ and Ω_p are some 'small subsets' of $\{1, \ldots, n\}$. With regard to the phase (for example) the corresponding SD model is the following (see Eq. (9)):

$$\rho^{(j)} + \lambda v^{(j)} + \alpha + \varepsilon^{(j)} = \left| \begin{array}{ll} \phi^{(j)} - \beta_{j_\phi} & \text{if } j \in \Omega_\phi \\ \phi^{(j)} & \text{otherwise} \end{array} \right.$$

The problem is to identify Ω_ϕ and Ω_p while getting least-squares estimates of the corresponding biases β_{j_ϕ} and β_{j_p}. The guiding idea is to the consider the contribution of these biases to w.

As $\Delta w = H\,\Delta y$ (see Eq. (61)), we must first see what is the contribution of these biases to y. At this level, the correction terms induced by e_{j_ϕ} and e_{j_p} are denoted by z_{j_ϕ} and z_{j_p}:

$$y \overset{\text{set}}{=} y - z_{j_\psi} \qquad z_{j_\phi} := \begin{bmatrix} e_{j_\phi} \\ 0 \end{bmatrix} \quad z_{j_p} := \begin{bmatrix} 0 \\ e_{j_p} \end{bmatrix} \qquad (64)$$

A notation such as $a \overset{\text{set}}{=} a + b$ means 'a is set equal to the current value of $a+b$.' The variations of w induced by e_{j_ϕ} and e_{j_p} are therefore characterized by the quantities f_{j_ϕ} and f_{j_p} defined below:

$$w \overset{\text{set}}{=} w - Hz_{j_\psi} \qquad f_{j_\phi} := Hz_{j_\phi} \qquad f_{j_p} := Hz_{j_p} \qquad (65)$$

As a result, the variation of w induced by the global bias z is characterized by the vector

$$Mz := \sum_{j_\phi \in \Omega_\phi} \beta_{j_\phi} f_{j_\phi} + \sum_{j_p \in \Omega_p} \beta_{j_p} f_{j_p} \qquad (66)$$

We are then led to solve, in the least-square sense, the equation $w - Mz$ '$=$' 0, in which the column vectors of M, the f_{j_ϕ}'s and f_{j_p}'s, have to be thoroughly selected. As clarified in Sect. 4.3, this operation is performed via a particular Gram-Schmidt orthogonalization process which is interrupted as soon as the corrected data are consistent with the model.

4.2 Global adaptation

Once the outlier sets Ω_ϕ and Ω_p have been identified, the model is to be updated consequently: A_i is completed by adding the columns associated with the corresponding bias variables β_{j_ϕ} and β_{j_p}. From Eqs. (42) and (64), these column matrices are respectively of the form

$$\begin{bmatrix} \mathcal{R}_\phi e_{j_\phi} \\ 0 \end{bmatrix} \qquad \begin{bmatrix} 0 \\ \mathcal{R}_p e_{j_p} \end{bmatrix} \qquad (67)$$

The global QR recursive process is then updated accordingly. The position variable, the SD biases and the float ambiguities are thus refined, as well as R_i and d_i in particular (see Fig. 3). When the QR process is initialized, or when the ambiguities are fixed, the SD biases provided by the adaptation process coincide with those provided by the identification procedure (see Sect. 4.1 and steps *2.4* & *2.5* in Sect. 4.3). The LS problem to be solved, which is then the same, is simply handled in a different manner.

4.3 Implementation

In the procedure described in this section (see the flow diagram shown in Fig. 6), we denote by Ω the set of identified outliers. At the beginning of this procedure, Ω is therefore empty: $\Omega := \Omega_\phi \cup \Omega_p = \emptyset$. For simplicity, we now restrict ourselves to the limit case defined in Sect. 1.4.2). We then set

$$|w|_{\max} = \max_{j_\psi \notin \Omega} |w_{j_\psi}| \qquad (68)$$

i.e. here: $|w|_{\max} = \max |w_{j_\psi}|$. Given some probability of false alarm θ_0, we define χ_0 as the upper $\theta_0/2$ probability point of the central normal distribution: $\chi_0 := N_{\theta_0/2}(0,1)$. For example, when θ_0 is equal to 0.001, χ_0 is of the order of 3.

1. Entrance local test

From Eqs. (59), (42) and (35), w is a reduced quantity. According to Eq. (39), in the absence of any bias, $|w|_{\max}$ must therefore be smaller than χ_0. In other terms, if $|w|_{\max} < \chi_0$, no outlier is to be searched: one then goes to step *4*. Conversely, if $|w|_{\max}$ is very large compared to χ_0 (say larger than 1000 for example), the QR process is to be reinitialized (see Sect. 3). In the other cases, the DIA procedure is initialized by setting $\mathfrak{r} = 1$ and $\Pi = \emptyset$; $\mathfrak{r}$ is a recursive index; the meaning of the auxillary set Π is defined in step *2.2* as soon as it begins to be built. At this stage, in the single-frequency case and in FLS mode (for example), the local redundancy is given by the formula $m = 2(n-1) - 3$.

2. Recursive identification of the outliers

2.1. Current set of potential outliers

Given some nonnegative constant $\kappa \leq 1$, form the current set of potential outliers (see Fig. 5):

$$\Pi_{\mathfrak{r}} := \left\{ j_\psi \notin \Omega : |w_{j_\psi}| \geq \kappa |w|_{\max} \right\}$$

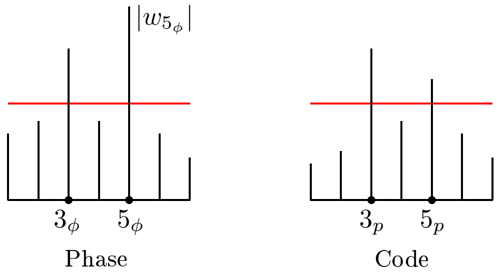


Fig. 5 *Notion of potential outliers in reduced mode.* The quantities $|w_{j_\psi}|$ shown here (in single-frequency mode) are the absolute values of the components of the (updated) residual w (see step *2.7*). In this illustration, $n = 7$, $\kappa = 0.5$ and $\Omega = \emptyset$; four potential outliers are identified: 3_ϕ, 5_ϕ, 3_p and 5_p. Here, the phase outlier 5_ϕ is likely to be the dominant potential outlier (see step *2.3*).

2.2. For each potential outlier $j_\psi \in \Pi_{\mathfrak{r}}$

Perform the following successive operations:

a) When $j_\psi \notin \Pi$, compute (see the context of Eqs. (64), (65), (61), (42) & (59) and Sect. 3.3)

$$f_{j_\psi} := H \cdot \left| \begin{array}{ll} z_{j_\phi} & \text{if } \psi = \phi \\ z_{j_p} & \text{if } \psi = p \end{array} \right.$$

Then, set

$$g_{j_\psi} := f_{j_\psi} \qquad \Pi \overset{\text{set}}{=} \begin{cases} \{j_\psi\} & \text{if } \Pi = \emptyset \\ \Pi \cup \{j_\psi\} & \text{otherwise} \end{cases}$$

By construction, Π is the set of potential outliers j_ψ for which f_{j_ψ} has already been computed.

b) If $\mathfrak{r} = 1$ go to step *2.2c*. Otherwise, at this level, $\{g^\circ_{\mathfrak{q}}\}_{\mathfrak{q}<\mathfrak{r}}$ is an orthonormal set. (This set is built, progressively, via step *2.4*.) Then, for each integer $\mathfrak{q} < \mathfrak{r}$, consider the inner product defined as follows:

$$\begin{aligned} \varsigma_{\mathfrak{q},j_\psi} &:= (g^\circ_{\mathfrak{q}} \cdot g_{j_\psi}) \\ &:= \sum_{\psi'=\phi,p} (g^\circ_{\mathfrak{q};\psi'} \cdot g_{j_\psi;\psi'}) \end{aligned}$$

This sum includes two terms. Depending on what ψ' refers to (ϕ or p), $g^\circ_{\mathfrak{q};\psi'}$ denotes the phase or code component of $g^\circ_{\mathfrak{q}}$, and likewise for $g_{j_\psi;\psi'}$. If $\varsigma_{\mathfrak{q},j_\psi}$ has not been computed yet, compute it, store it in memory, and perform the Gram-Schmidt orthogonalization operation

$$g_{j_\psi} \overset{\text{set}}{=} g_{j_\psi} - \varsigma_{\mathfrak{q},j_\psi} g^\circ_{\mathfrak{q}}$$

By construction, $\varsigma_{\mathfrak{q},j_\psi} = (g^\circ_{\mathfrak{q}} \cdot f_{j_\psi})$. At the end of all these operations, g_{j_ψ} is orthogonal to $g^\circ_{\mathfrak{q}}$ for any $\mathfrak{q} < \mathfrak{r}$.

c) Consider the projection of w on the one-dimensional space generated by g_{j_ψ}, i.e., $(h_{j_\psi} \cdot w) h_{j_\psi}$ where $h_{j_\psi} := g_{j_\psi} / \|g_{j_\psi}\|$. The norm of this projection is equal to $|(h_{j_\psi} \cdot w)|$, the absolute value of the quantity

$$\gamma_{j_\psi} := (g_{j_\psi} \cdot w) / \varrho_{j_\psi} \qquad \varrho_{j_\psi} := \|g_{j_\psi}\|$$

Explicitly,

$$\begin{aligned} (g_{j_\psi} \cdot w) &:= \sum_{\psi'=\phi,p} (g_{j_\psi;\psi'} \cdot w_{\psi'}) \\ \|g_{j_\psi}\|^2 &:= \sum_{\psi'=\phi,p} \|g_{j_\psi;\psi'}\|^2 \end{aligned}$$

2.3. Dominant potential outlier

The identified outlier $\bar{\jmath}_{\bar{\psi}}$ is defined as the dominant potential outlier, i.e., the potential outlier for which $|\gamma_{j_\psi}|$ is maximal:

$$\bar{\jmath}_{\bar{\psi}} := \underset{j_\psi \in \Pi_{\mathfrak{r}}}{\text{arg max}} |\gamma_{j_\psi}|$$

We then set

$$\omega_{\mathfrak{r}} := \bar{\jmath}_{\bar{\psi}} \qquad \Omega \overset{\text{set}}{=} \begin{cases} \{\omega_{\mathfrak{r}}\} & \text{if } \mathfrak{r} = 1 \\ \Omega \cup \{\omega_{\mathfrak{r}}\} & \text{if } \mathfrak{r} > 1 \end{cases}$$

$$\gamma^\circ_{\mathfrak{r}} := \gamma_{\omega_{\mathfrak{r}}} \qquad g^\circ_{\mathfrak{r}} := g_{\omega_{\mathfrak{r}}} / \varrho_{\omega_{\mathfrak{r}}}$$

Superscript $\circ$ stands for omega (and outlier). At this level, Ω is the current set of identified outliers:

$$\Omega = \{\omega_{\mathfrak{q}}\}_{\mathfrak{q}=1}^{\mathfrak{r}}$$

By construction, $\{g^\circ_{\mathfrak{q}}\}_{\mathfrak{q}=1}^{\mathfrak{r}}$ is an orthonormal basis of the current range of M; $\sum_{\mathfrak{q}=1}^{\mathfrak{r}} \gamma^\circ_{\mathfrak{q}} g^\circ_{\mathfrak{q}}$ is the projection of w on this space. With regard to Eq. (66), we then set

$$\beta^\circ_{\mathfrak{r}} := \beta_{\omega_{\mathfrak{r}}} \qquad f^\circ_{\mathfrak{r}} := f_{\omega_{\mathfrak{r}}}$$

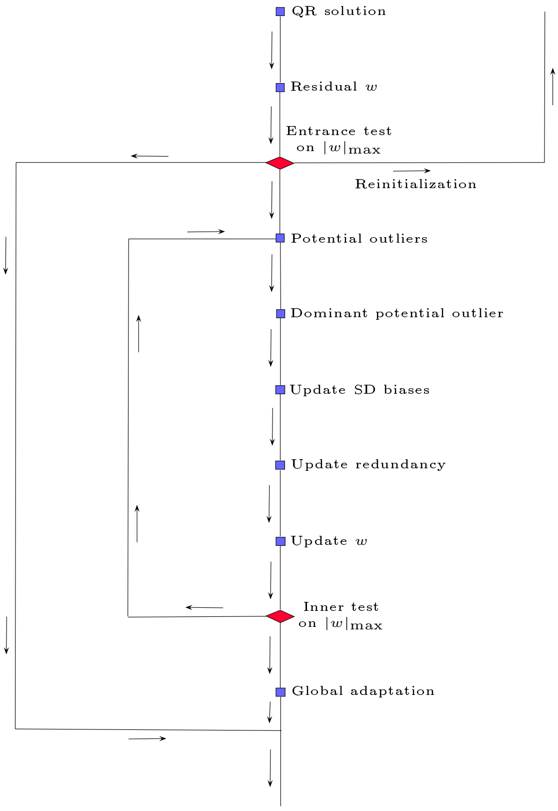


Fig. 6 *Flow diagram of the DIA procedure in reduced mode.* At each step of the identification process, the (updated) residual w is analyzed on the grounds of Eq. (68): see steps *1*, *2.7* and *2.8*. This allows the potential outliers to be selected (see Fig. 5). The outliers can thus be identified, in a recursive manner, via a particular orthogonalization Gram-Schmidt process. This QR Gram-Schmidt process also provides the SD biases, and thereby the cycle slips if any. When the ambiguity are not fixed, these biases are slightly refined through the global adaptation process described in Sect. 4.2.

2.4. Components of $g^\circ_\mathfrak{r}$ in the basis of the $f^\circ_\mathfrak{q}$'s

These components are denoted by $\mathrm{u}_{\mathfrak{q},\mathfrak{r}}$:

$$g^\circ_\mathfrak{r} = \sum_{\mathfrak{q}=1}^{\mathfrak{r}} \mathrm{u}_{\mathfrak{q},\mathfrak{r}} f^\circ_\mathfrak{q}$$

They are computed via the QR Gram-Schmidt formulas (see, e.g., Björck 1996)

$$\mathrm{u}_{\mathfrak{q},\mathfrak{r}} = \begin{cases} -\dfrac{1}{\varrho_{\omega_\mathfrak{r}}} \displaystyle\sum_{\mathfrak{q} \leq \mathfrak{q}' < \mathfrak{r}} \mathrm{u}_{\mathfrak{q},\mathfrak{q}'}\, \varsigma_{\mathfrak{q}',\omega_\mathfrak{r}} & \text{if } \mathfrak{q} < \mathfrak{r} \\[2ex] \dfrac{1}{\varrho_{\omega_\mathfrak{r}}} & \text{if } \mathfrak{q} = \mathfrak{r} \end{cases}$$

for $1 \leq \mathfrak{q} \leq \mathfrak{r}$. The $\mathrm{u}_{\mathfrak{q},\mathfrak{r}}$'s are the entries of the $\mathfrak{r}^{\text{th}}$ column of an upper triangular matrix U.

2.5. Update the SD biases

According to Eq. (66), the SD biases $\beta^\circ_\mathfrak{q}$ are the components of $\sum_{\mathfrak{q}=1}^{\mathfrak{r}} \gamma^\circ_\mathfrak{q} g^\circ_\mathfrak{q}$ in the basis of the $f^\circ_\mathfrak{q}$'s:

$$\sum_{\mathfrak{q}=1}^{\mathfrak{r}} \gamma^\circ_\mathfrak{q} g^\circ_\mathfrak{q} = \sum_{\mathfrak{q}=1}^{\mathfrak{r}} \beta^\circ_\mathfrak{q} f^\circ_\mathfrak{q}$$

Denoting by $[\gamma^\circ]$ the column matrix with entries $\gamma^\circ_\mathfrak{q}$ (from $\mathfrak{q} = 1$ to $\mathfrak{r}$), and likewise for $[\beta^\circ]$, we have

$$[\beta^\circ] = \mathrm{U}[\gamma^\circ]$$

The SD biases are therefore to be updated as follows:

$$\beta^\circ_\mathfrak{q} \overset{\text{set}}{=} \begin{cases} \beta^\circ_\mathfrak{q} + \mathrm{u}_{\mathfrak{q},\mathfrak{r}}\gamma^\circ_\mathfrak{r} & \text{if } \mathfrak{q} < \mathfrak{r} \\ \mathrm{u}_{\mathfrak{r},\mathfrak{r}}\gamma^\circ_\mathfrak{r} & \text{if } \mathfrak{q} = \mathfrak{r} \end{cases} \qquad (\text{for } 1 \leq \mathfrak{q} \leq \mathfrak{r})$$

2.6. Update the local redundancy

$$m \overset{\text{set}}{=} m - 1$$

If $m = 0$ go to step *3*.

2.7. Update w and $|w|_{\max}$

$$w \overset{\text{set}}{=} w - \gamma^\circ_\mathfrak{r} g^\circ_\mathfrak{r} \qquad |w|_{\max} \overset{\text{set}}{=} \max_{j_\psi \notin \Omega} |w_{j_\psi}|$$

2.8. Inner local test

If $|w|_{\max} > \chi_0$, update the recursive index: $\mathfrak{r} \overset{\text{set}}{=} \mathfrak{r} + 1$. Then, go to step *2*.

3. Global adaptation

Update the global QR recursive process by taking account of the identified bias variables (see Sect. 4.2).

4. End

5 Examples

The QR implementation presented in this paper was validated by processing two GPS-data sets in dual-frequency mode (L1-C/A, L2-P). Shortly, these sets correspond to the following cases:

- **Static case.** Static reference receiver; static user receiver; 4907 epochs at 1 Hz; baseline size of the order of 250 m.
- **Kinematic case.** Static reference receiver; mobile user's car receiver; 973 epochs at 2 Hz; maximal baseline size of the order of 850 m.

The static case was studied to check our programs. In both cases, the standard deviations σ_Φ and σ_p were of the order of 3 mm and 55 cm, respectively (see Eq. (36)).

The reduced data were therefore centralized differences of type (39) with $\eta_j = 2$ for all j; χ_0 was set equal to 3. These data were processed in forced RLS mode (with initializations in LS mode).

As illustrated in Eq. (43), the float ambiguities were put in reverse order. Furthermore, to benefit from the analysis presented in Sects. 3.4 and 3.5, the L1 and L2 ambiguities were interwoven, as well as the L1 and L2 data in their phase and code column submatrices.

The optimal and suboptimal ambiguity solutions, $\dot{v}$ and $\dot{\dot{v}}$ respectively, were obtained (at each RLS epoch) by solving the nearest-lattice point problem defined in Sect. 3.6. It was thus possible to control the value of the 'global ambiguity-resolution parameter'

$$\varrho_1 := \frac{\|\dot{v} - \hat{v}\|^2_{V_{\hat{v}}^{-1}}}{\|\dot{\dot{v}} - \hat{v}\|^2_{V_{\hat{v}}^{-1}}} \tag{69}$$

The 'local ambiguity-resolution parameter'

$$\varrho_2 := \frac{|\dot{w}|_{\max}}{|\dot{\dot{w}}|_{\max}} \tag{70}$$

was also computed. Here, $\dot{w}$ and $\dot{\dot{w}}$ denote the values of the optimal and suboptimal residuals, respectively; note that the bias variables are then included in the local variable u_i. When

$$\varrho_1 \lesssim 0.5 \quad \text{or} \quad \varrho_2 \lesssim 0.4 \quad \text{(validation criterion)} \tag{71}$$

the ambiguities can be regarded as fixed.

All the programs were written in C language, including the LLL algorithm and the nearest-lattice point section. The first data set of 4907 epochs was thus processed, with $\kappa = 0$, in about five seconds on a standard personal computer. With $\kappa = 1$, this CPU time was reduced to three seconds with exactly the same results. The second data set of 973 epochs was processed in about two seconds for $\kappa = 0$, and in about one second for $\kappa = 1$.

5.1 Static case

In this case, due to major data-frame problems, the process was reinitialized at the following epochs: 1301, 3010 and 4689. As specified below for the first run, the ambiguities were fixed immediately. The position of the user receiver was thus retrieved, up to one or two centimeters, except for the initialization epochs of the four runs to be considered: 1, 1301, 3010 and 4689 (see Fig. 7).

We now concentrate on the first run. Seven or eight satellites were then visible: satellites 2, 5, 7, 8, 9, 23, 26 and sometimes 21. The latter appears and disappears (in an alternate manner) at the following epochs: 365, 878, 883, 884, 887, 888, 892, 896, 911, 936, 1004, 1098, 1130.

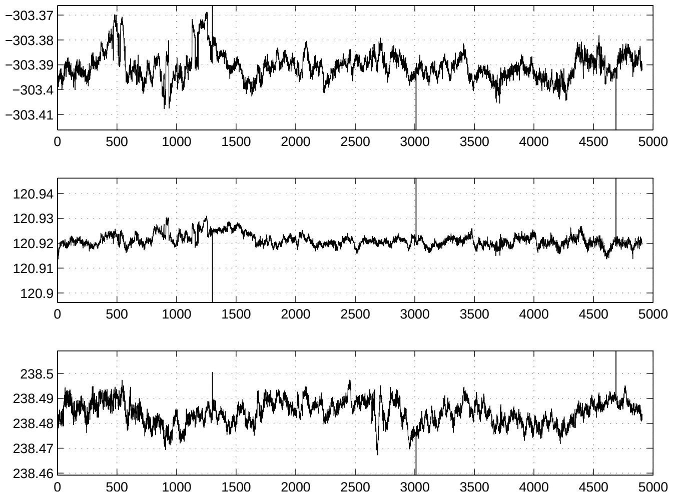


Fig. 7 *Static case (4907 epochs).* Relative coordinates (expressed in meters) of the user and reference receivers in the Earth-centred Earth-fixed (ECEF) frame: x, y, z (from the top to the bottom); see text and Fig. 8.

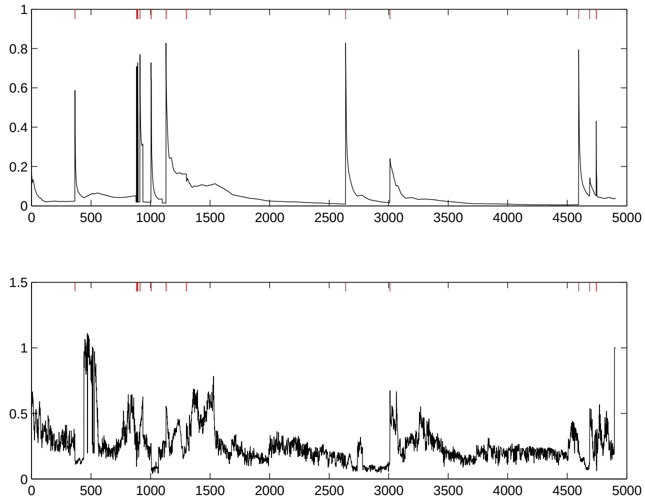


Fig. 8 *Static case (4907 epochs).* Ambiguity resolution parameters ϱ_1 (at the top) and ϱ_2 (at the bottom); see the context of Eqs. (69) to (71). The ambiguities are fixed, except at the initialization epoch 1 and at the reinitialization epochs 1301, 3010 and 4689 (see Fig. 7 and the corresponding red ticks). The other red ticks correspond to the epochs where a new satellite appears or reappears.

At epoch 1 (in LS mode), a code bias was identified on satellite 2 at frequency f_1; see steps *2.4* and *2.5* in Sect. 4.3. Its value, 7.02 m, was of course the same as that found by the adaptation process; see Sect. 4.2 and Fig. 6. The data of epoch 2 were of course processed in RLS mode. Again, a code bias was identified on satellite 2. As expected, its value, 6.70 m, was very close to that provided by the global adaptation process: 6.77 m. The ambiguities proved then to be fixed (see Table 1): ϱ_1 was smaller than 0.16 with ϱ_2 smaller than 0.65 (see Fig. 8 and Eqs. (69) to (71)). The code bias thus found was 5.42 m. Here, $|\dot{w}|_{\max} = 3.22$ and $|\dot{\dot{w}}|_{\max} = 5.07$.

Table 1: *Static case. Dual-frequency DD ambiguities.* The ambiguities shown here were fixed at epoch 2, just after the initialization epoch (see text).

satellite	f_1	f_2
2	0	0
5	995 532	783 561
7	1 585 927	329 961
8	−1 542 232	−893 259
9	13 115 987	10 232 032
23	6 934 437	4 872 157
26	10 017 404	7 778 866

As soon as satellite 21 appeared (at epoch 365), the corresponding ambiguities were immediately fixed:

satellite	f_1	f_2
21	−1 632 504	−777 230

At epoch 1093, large phase biases were identified on the L2 and L1 SD phase data of that satellite: 0.143 m and 0.107 m, respectively. As shown by the results obtained at the next epoch, these biases announced effective cycle slips. Indeed, at epoch 1094, one cycle slip was identified on the L2 SD phase of satellite 21, and likewise for the L1 SD phase of that satellite. More precisely, the biases identified by the RLS DIA procedure were then the following:

$$\begin{aligned}\beta_{f_2,21_\phi} &= \ \ 0.227\,\mathrm{m} \simeq \lambda_2\\ \beta_{f_1,21_\phi} &= \ \ 0.195\,\mathrm{m} \simeq \lambda_1\\ \beta_{f_1,21_p} &= -4.861\,\mathrm{m}\\ \beta_{f_2,21_p} &= \ \ 3.974\,\mathrm{m}\end{aligned}$$

At that epoch, the entrance value of $|w|_{\max}$ was large compare to 3 : 28.40. The outliers were then identified as specified below:

Outlier	$\lvert w\rvert_{\max}$
$(f_2\,;21_\phi)$	29.64
$(f_1\,;21_\phi)$	5.49
$(f_1\,;21_p)$	4.49
$(f_2\,;21_p)$	2.30

Here, the value in the right-hand side column is the corresponding residual value of $|w|_{\max}$. Corrected from the cycles slips thus identified, the data were then processed without any large phase biases until the disappearance of satellite 21 at epoch 1098, and then without any difficulty until the major data-frame problem at epoch 1301.

In the second run, from epoch 1301 to epoch 2060 included, all the previous 8 satellites were visible. The reference satellite s_1 (satellite 2) then disappeared at epoch 2061. A similar situation occured in the fourth run with nine satellites: the reference satellite s_1 (satellite 1 in that run) disappeared at epoch 4743. To check the section of the program corresponding to the disappearance of other satellites in RLS mode (see Sect. 3.5), the SD data of satellite s_2 (then satellite 5) were discarded at epoch 4775. As expected, the corresponding results were correct.

From epoch 4897 to the end of the fourth run, the optimal and suboptimal sets of L1 ambiguities coincide up to an integer constant: the unity for all j; the optimal and suboptimal sets of L2 ambiguities are then identical. As at those epochs, the reference satellite is not visible, the reduced values of $\dot{v}$ and $\dot{\dot{v}}$ are the same (see Eq. (53) and Eqs. (39) & (40) with $\eta_j = 2$ for all j). It then follows that $\dot{w} = \dot{\dot{w}}$, hence $\varrho_2 = 1$ (see Fig. 8). The ambiguities are however fixed. Indeed ϱ_1 is then less than 0.04 (see Eq. (71)).

5.2 Kinematic case

In this case, nine to eleven satellites were visible: satellites 4, 9, 16, 18, 19, 22, 23, 24, 28, 29 and 32. The ambiguities were immediately fixed with ϱ_1 less than 0.15 and ϱ_2 less than 0.33 (see Table 2 and Figs. 9 & 10; satellite 9 was not then visible).

Table 2: *Kinematic case. Dual-frequency DD ambiguities.* The ambiguities shown here were fixed at epoch 2, just after the initialization epoch (see text).

satellite	f_1	f_2
4	0	0
16	−577 343	−425 713
18	−489 386	−357 110
19	16 040	40 057
22	187 137	178 615
23	−611 408	−448 519
24	−188 663	−122 172
28	−1 651 396	−1 238 734
29	363 726	308 953
32	−19 687	2 051

A major data problem appeared at epoch 222. The process was then reinitialized by the RLS DIA procedure. Indeed, the entrance value of $|w|_{\max}$ was greater than 10^6 (see step *1* in Sect. 4.3). The ambiguities were then fixed again, but only eleven seconds later (after epoch 244; see Fig. 10 and Eq. (71)).

Just to show the efficiency of our approach, cycles slips were imposed at epoch 960: −1 cycle in the reception of the f_1-signal coming from the reference satellite; 2 cycles in the reception of the f_2-signal coming from satellite 23; 1 cycle in the reception of the f_2-signal coming from satellite 29.

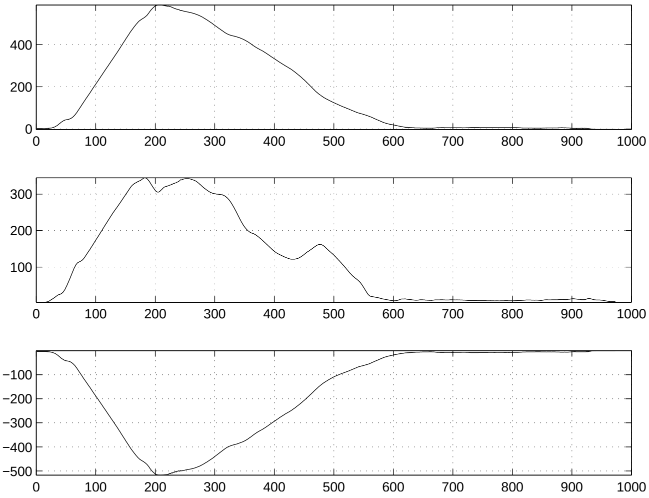


Fig. 9 *Kinematic case (973 epochs).* Relative positions (in meters) of the user and reference receivers in the ECEF frame: x, y, z (from the top to the bottom); see text and Fig. 10.

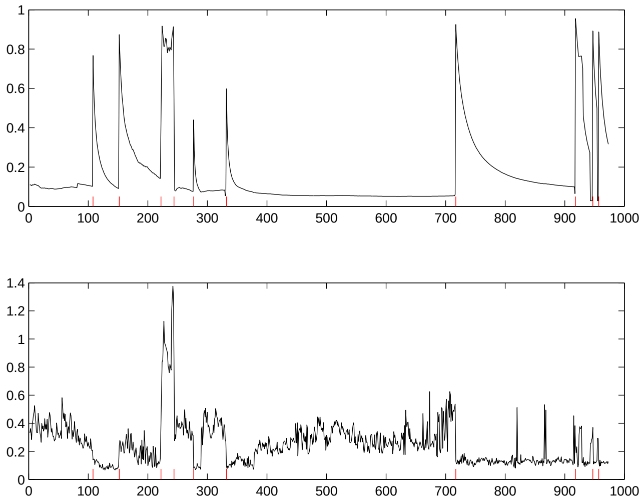


Fig. 10 *Kinematic situation (973 epochs).* Ambiguity resolution parameters ϱ_1 (at the top) and ϱ_2 (at the bottom); see the context of Eqs. (69) to (71). The ambiguities are fixed, except at the initialization epoch and from epochs 222 to 244 included (see text and the corresponding red ticks). The other red ticks correspond to the epochs where a new satellite appears or reappears.

At that epoch, the entrance value of $|w|_{\max}$ was then of the order of 69. In the RLS DIA procedure, the outliers were then identified as follows:

Outlier	$\lvert w\rvert_{\max}$
$(f_2\,;23_\phi)$	46.57
$(f_2\,;29_\phi)$	29.06
$(f_1, 4_\phi)$	2.66

The SD biases finally obtained by the process were then the following:

$$\begin{aligned}\beta_{f_2,23_\phi} &= 0.488\,\text{m} \simeq 2\lambda_2\\ \beta_{f_2\,;29_\phi} &= 0.239\,\text{m} \simeq \lambda_2\\ \beta_{f_1\,;\,4_\phi} &= -0.196\,\text{m} \simeq -\lambda_1\end{aligned}$$

Corrected from the cycles slips thus identified, the data were processed without any difficulty until the end of the run (epoch 973).

6 Concluding comments

As clarified in Sect. 1.4, the notions of reduction and centralization correspond to the same concept. The variance-covariance matrix of the reduced or centralized data is the identity. For example, in the single-baseline case, the reference formulas are Eqs. (39) and (40). In the centralized approaches, the QR method can therefore be applied directly. This not the case in the usual DD approach. Indeed, the Cholesky factorization of the inverse of the variance-covariance matrix of the DD data must then be performed. Moreover, in the centralized approaches, all the SD data are handled in the same manner. The corresponding numerical codes are therefore more readable than those of their DD versions.

The QR implementation of GNSS centralized approaches is also well suited to quality control. The search for the potential outliers is performed by simple inspection of the absolute value of the components of the successive updated residuals (see Fig. 5 and step *2.7* in Sect. 4.3). The statistical tests are thereby very simple (see steps *1* and *2.8* in Sect. 4.3). Moreover, as the Givens rotations of the QR recursive processes can easily be stored in memory, the variational calculations involved in the DIA method can be performed in a very efficient manner; see Sect. 3.3 and step *2.2* in Sect. 4.3. Furthermore, the QR global adaptation step of the DIA method nicely completes the QR Gram-Schmidt step *2.4* of the local identification process described in Sect. 4.3. The SD biases, among which the cycles slips (if any), are thus determined in two different ways.

For simplicity, the study presented in this paper was restricted to the case of RTK observations with a single baseline of local scale. The extension to multiple-baseline networks with possibly missing data follows the guidelines of the present contribution. The main points to be developed concern the following topics:

- Handling the integer ambiguities;
- Reduction of the undifferential optimization problem (equivalent of Sect. 1.4 for the undifferential data);
- QR solution of the reduced optimization problem;
- Integer-ambiguity resolution;
- Identifiable biases;
- Related DIA method.

References

Agrell E., Eriksson T., Vardy A. and Zeger K. (2002) ***Closest point search in lattices***. IEEE Trans. Inform. Theory. 48: 2201–2214.

Bierman G.J. (1977) ***Factorization methods for discrete sequential estimation***, Vol. 128 in Mathematics in science and engineering, Academic Press, Inc. New-York.

Björck A. (1996) ***Numerical methods for least-squares problems***, SIAM.

Chang X.-W. and Guo Y. (2005) ***Huber's estimation in relative GPS positioning: computational aspects***. J. Geod. 79: 351–362.

Feng Y. and Li B. (2008) ***Three-carrier ambiguity resolution: generalized problems, models, methods and performance analysis using semi-generated triple frequency GPS data***. Proc. ION GNSS-2008. Savannah, Georgia USA: 2831-2840

Golub G.H. and van Loan C.F. (1989) ***Matrix computations***, second edition, The Johns Hopkins University Press, Baltimore, Maryland.

Hewitson S., Lee H.K. and Wang J. (2004) ***Localizability analysis for GPS/Galileo receiver autonomous integrity monitoring***. The Journal of Navigation, Royal Institute of Navigation 57: 245–259.

Lannes A. (2007a) ***On the concept of reduced difference in differential GNSS***. C. R. Mécanique. 335: 720–726.

Lannes A. (2007b) ***Differential GPS: the reduced difference approach***. J. GPS. 6: 23–37.

Lannes A. (2008) ***GNSS networks with missing data: identifiable biases and potential outliers***. Proc. ENC GNSS-2008. Toulouse, France: 1–11.

Liu X. (2002) ***A comparison of stochastic models for GPS single differential kinematic positioning***. Proc. ION GPS-2002. Portland, Oregon USA: 1830-1841.

Loehnert E., Wolf R., Pielmeier J., Werner W. and Zink T. (2000) ***Concepts and performance results on the combination of different integrity methods using UAIM and GNSS without SA***. Proc. ION GPSS-2000. Salt Lake City, Utah USA: 2831-2840

Luk F.T. and Tracy D.M. (2008) ***An improved LLL algorithm***. Linear algebra and its applications. 428: 441–452.

Shi P. H. and Han S. (1992) ***Centralized undifferential method for GPS network adjustment***. Australian Journal of Geodesy, Photogrammetry and Surveying. 57: 89-100.

Strang G. and Borre K. (1997) ***Linear algebra, geodesy, and GPS***, Wellesley-Cambridge Press, Massachussets.

Teunissen P.J.G. (1990) ***An integrity and quality control procedure for use in multi sensor integration***. Proc. ION GPS-90. Colorado Springs, Colorado USA: 513-522

Tiberius C.C.J.M. (1998) ***Recursive data processing for kinematic GPS surveying***, Publications on Geodesy, New series: ISSN 0165 1706, Number 45, Netherlands Geodetic Commission, Delft.

Xu P. (2001) ***Random simulation and GPS decorrelation***. J. Geod. 75: 408–423.

Application of Running Average Function to Non-Dispersive Errors of Network-Based Real-Time Kinematic Positioning

Samsung Lim and Chris Rizos
School of Surveying and Spatial Information Systems, The University of New South Wales, Sydney, Australia.

Tajul Musa
Department of Geomatics Engineering, Universiti Teknologi Malaysia, Malaysia.

Abstract

The GPS errors can be separated into a frequency-dependent or dispersive component (e.g. the ionospheric delay) and a non-dispersive component (e.g. the tropospheric delay and orbit biases). Dispersive and non-dispersive errors have different dynamic effects on the GPS network corrections. The former exhibits rapid changes with high variations due to the effect of free electrons in the ionosphere, whilst the latter change slowly and smoothly over time due to the characteristic behaviour of the tropospheric delay and the nature of orbit biases. It is found that the non-dispersive correction can be used to obtain better ionosphere-free measurements, and therefore helpful in resolving the long-range integer ambiguity of the GPS carrier-phase measurements. A running average is proposed in this paper to provide a stable network correction for the non-dispersive term. Once the integer ambiguities have been resolved, both dispersive and non-dispersive corrections can be applied to the fixed carrier-phase measurements for positioning step so as to improve the accuracy of the estimated coordinates. Instantaneous positioning i.e. single-epoch positioning, has been tested for two regional networks: SydNET, Sydney, and SIMRSN, Singapore. The test results have shown that the proposed strategy performs well in generating the network corrections, fixing ambiguities and computing a user's position.

Keywords: GPS, RTK, network-RTK, running average.

1. Introduction

Real-time kinematic (RTK) ambiguity resolution, a key step for precise GPS positioning, is complicated due to many error sources in the carrier-phase measurements. These errors can be grouped into station- and distance-dependent errors. Station-dependent errors such as receiver-biased errors, multipath effects and measurement noises, notably degrade the ambiguity resolution. The effort to reduce this type of errors has been considerably undertaken in the past decade. Ambiguity resolution is also seriously affected by the presence of the distance-dependent errors: ionospheric delay, tropospheric delay and orbit biases. Due to the distance-dependent errors, reliable RTK ambiguity resolution is limited to relatively short inter-receiver distances, typically of the order of 10km or 25km at the maximum. However, there exists a strong demand to extend the baseline length, without sacrificing RTK performance. The use of multiple GPS reference stations i.e. a GPS network, makes it possible.

GPS networks have been deployed for many years, providing opportunities to mitigate distance-dependent errors in many ways. A good example is the network of the International GNSS Service (IGS), and its products (cf. http://igscb.jpl.nasa.gov). To date the coverage of IGS is not dense enough to be sensitive to small-scale errors, and therefore does not meet the requirement of regional or local GPS users. Although the IGS products are improving, many countries have developed their own regional or local GPS networks. The inter-station distances in these networks are kept below 200km in order to model the distance-dependent errors adequately.

The concept and the technique of carrier-phase network-based RTK positioning were introduced by Wanninger (1995), based initially on utilising three reference stations of a GPS network. Estimated distance-dependent errors for each reference station are combined in order to interpolate and estimate the same types of errors for users within the network coverage. A variety of algorithms for estimating such 'network corrections' exist, but the popular algorithms are: Virtual Reference Station (VRS) implemented by Trimble (Lynn & Anil, 1995; Wanninger, 1997) and Area Correction Parameters which is also known as Flächen Korrektur Parameter (FKP) in German (Wubbena & Bagge, 1998). The major difference between the two methods is that they used different approaches to make corrections for the rovers. VRS provides a rover with "error-removed" data for a virtual reference station in close proximity to the rover. Therefore the rover must send its approximate position to VRS. FKP is suitable for broadcasting the corrections to multiple users in the network because the FKP corrections are actually interpolation coefficients for the rovers' position. Previous work has shown that the network-based technique is an efficient means of improving long-range ambiguity resolution, and enables high accuracy positioning with less dense GPS reference station networks than would be the case if single-baseline RTK techniques were used.

The network corrections can be separated into dispersive (ionosphere-related) and non-dispersive (troposphere- and orbit-related) components according to their dependency on GPS signal frequency. Euler *et al.* (2004) discussed the impact of incorrectly determined network integer ambiguity on the separated dispersive and non-dispersive corrections. Keenan *et al.* (2002) proposed a user standard correction transmission format that separates the network corrections. Dispersive and non-dispersive components have different dynamic effects. Typically dispersive components exhibit rapid changes, with high variations due to the effect of free electrons in the ionosphere (Hernandes *et. al*, 1999; Odijk, 2002). On the other hand, non-dispersive components change slowly and smoothly over time due to the characteristic behaviour of the tropospheric delay and the nature of orbit biases (Tajul *et al.,* 2005). Further attention should be given to the separation, and the dynamic effect, of the network corrections.

In this paper, a running average function is proposed to improve non-dispersive corrections. In order to validate this proposition, tests of instantaneous ambiguity resolutions are conducted and compared with conventional network-RTK positioning. Test results with and without applying the function will be compared.

2. Methodology of Network-RTK

Network-RTK needs all GPS reference stations to transmit their raw GPS measurements to a control centre. The network algorithm at the control centre will select one of them as a master station and calculate the network corrections. Then the network corrections need to be distributed to users.

Because of the long distances between the stations in the network, the task of the network ambiguity resolution is challenging. Furthermore, the process needs to be done in real-time. Several discussions about this process can be found in Hu *et al.* (2005), Chen *et al.* (2004), Dai (2002) and Odijk (2002). For the static mode, the ambiguity resolution process can take advantage of long observation sessions. In the real-time mode, however, the degree of freedom is less. Hence, all measurement errors need to be appropriately modelled, and a fast ambiguity search and validation methodology is required.

To assist network ambiguity resolution, the data from dual-frequency receivers are processed, choke-ring type antennas are used, as well as knowledge of the network baseline lengths and precise (predicted) ultra-rapid orbits from the IGS, low multipath environment is assumed, and the reference stations are static. The processing takes advantage of various linear combinations of carrier-phase and pseudorange measurements. Well-known linear combinations, such as the widelane and the ionosphere-free, are often used for network ambiguity resolution (Han, 1997; Sun *et al.*, 1999).

Once network ambiguities are fixed, the residuals are used to approximate the distance-dependent errors within the area. This "lump sum" approach is applied in order not to combine the residuals into a single network correction; hence they are separated according to whether they are dispersive or non-dispersive. The separation can be easily done via geometry-free and ionosphere-free combinations. Properties of these combinations can be found in Rizos (1997). The next step is to interpolate these residuals relative to the user's approximate position, which in turn provides the user with the network correction. Dai (2002) discussed several interpolation methods that can be used for this purpose. A linear interpolation algorithm is adequate to perform this task for a local network. Therefore the linear combination method (LCM) (Han, 1997) is used in this study.

Due to the rapid changes and high variability of the ionosphere effect, interpolating the dispersive component has to be performed as frequently as possible (e.g. epoch-by-epoch). Conversely, rapid variations can be observed in the non-dispersive component because of remaining multipath and noises in the ionosphere-free measurements. Hence, a similar attempt to interpolate this component, as in the case of dispersive component, will have a tendency of increasing residuals. For this reason it is suggested in this paper that non-dispersive errors should not be interpolated on an epoch-by-epoch basis. In addition, a running average is applied to non-dispersive errors in order to obtain smooth non-dispersive correction. This smoothed result remains valid for many epochs (say 5 to 10 minutes) and the process should be continuously running for the next 'windows'.

3. Tests for Local GPS Networks

Two local GPS networks in different geographical locations were tested in this study. One is the Sydney Network (SydNET) located in the mid-latitudes (latitude range 33°36' – 34°08'S, and longitude range 150°34' – 151°12'E), and the other is the Singapore Integrated

Multiple Reference Station Network (SIMRSN) located near the equator (latitudes 1°15' – 1°30'N, and longitudes 103°40' – 103°59'E). It is expected that atmospheric effects are more severe in the equatorial area. Figures 1 and 2 show the locations of the stations within SydNET and SIMRSN, respectively.

To investigate the proposed network processing strategy, tests were conducted in post-processed, but 'simulated' RTK mode. For verification purposes, the data have been processed in static mode. Stations SPWD of SydNET and LOYA of SIMRSN were selected as the two networks' master stations. Meanwhile, the station VILL of SydNET and NYPC of SIMRSN were treated as user stations. The selection is made to avoid severe multipath for the user station because the proposed network algorithm is not aimed at mitigating such effects at the moment. Other stations were considered to be reference stations (see Figures 1 and 2). It was assumed that the two networks had access to IGS ultra-rapid orbit data and were equipped with data transmission facilities. Reductions to the user's and the master's raw GPS measurements by the network correction were avoided in the first place, except for an *a priori* tropospheric model. The network correction (i.e. dispersive and non-dispersive terms) was generated by removing satellites in the master-to-reference combinations whose elevations were less than 10°. For master-to-user processing, it was further categorised by changing the satellites' cut-off elevation angles from 10° to 15° and 20°.

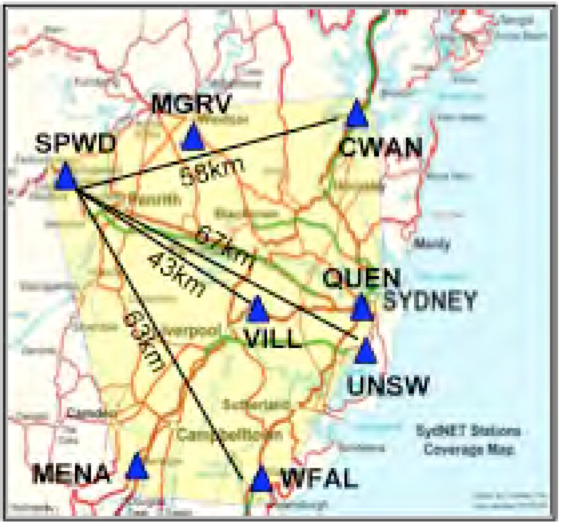


Fig. 1 SydNET network

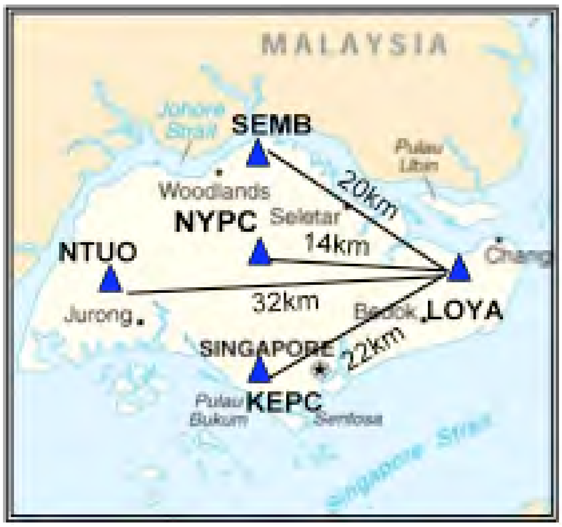


Fig. 2 SIMRSN network

4. Test Results and Analysis

Figures 3 (SydNET) and 4 (SIMRSN) show the original master-to-user double-differenced residuals of dispersive and non-dispersive effects for all satellite combinations. Associated network corrections are also highlighted in these figures.

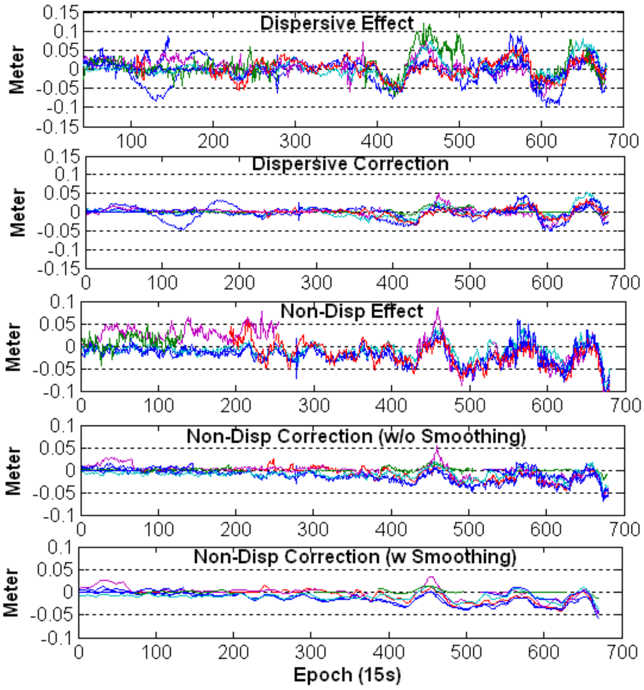


Fig. 3 SydNET Test. Top two: residuals of DD dispersive effect (top) and dispersive correction (bottom). Bottom three: residuals of DD non-dispersive effect (top), original correction (middle) and smoothed correction (bottom) for non-dispersive. Baseline: SPWD-VILL (~43km) in Sydney. Day of Year (DoY): 131/05 and the observation period of 3hrs (10.00pm-1.00am, local time)

As can be seen in Figure 3, both dispersive and non-dispersive corrections have performed reasonably well. The magnitude of the corrections is approximately almost the same or half the magnitude of the original residuals. Inspecting the residual patterns, it is obvious that the network corrections exhibit some trends. In Figure 4, however, there are less accurate corrections even though the baseline length in this network is shorter. This can be explained by the stronger atmospheric activity in the equatorial region. Therefore, this complicates the master-to-reference ambiguity resolution, which in turn results in lower quality network corrections.

The non-dispersive correction performed well in both tests when the smoothing function is applied. The magnitudes and trends of the smoothed corrections are in the range of the non-dispersive residuals. It can be noticed from both figures that network corrections for some epochs are not available, especially for low elevation satellites. Figures 5 and 6 indicate the number of satellites in view and the available corrections for the VILL and NYPC stations.

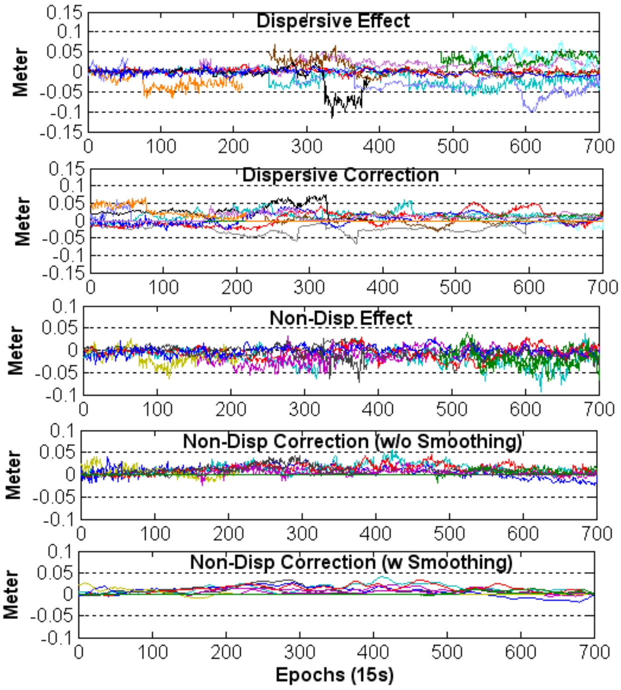


Fig. 4 SIMRSN Test. Top two: residuals of DD dispersive effect (top) and dispersive correction (bottom). Bottom three: residuals of DD non-dispersive effect (top), original correction (middle) and smoothed correction (bottom) for non-dispersive. Baseline: LOYA-NYPC (~14km) in Singapore. DoY: 166/03 and the observation period of 3hrs (8.00am-11.00am, local time)

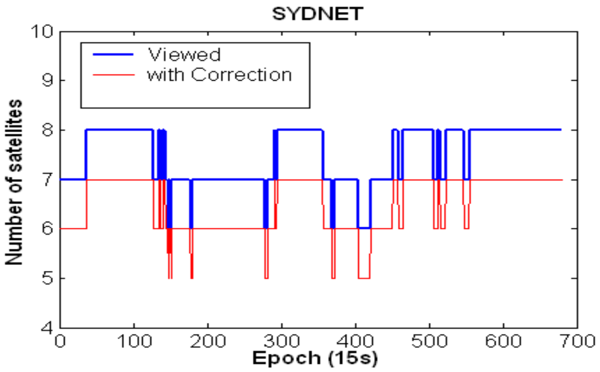


Fig. 5 Number of satellites in view (at 10 º elevations and above) and available corrections for the station VILL in SydNET

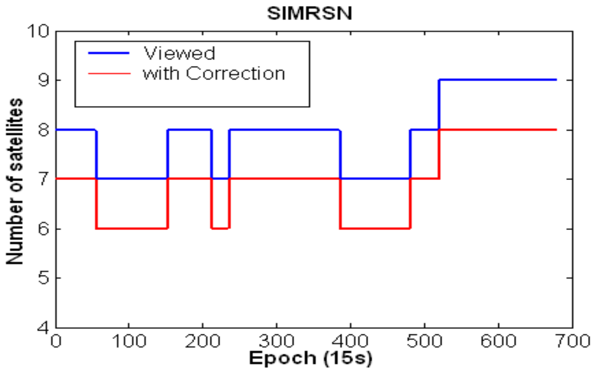


Fig. 6 Number of satellites in view (at 10 º elevations and above) and available corrections for the station NYPC in SIMRSN

During the period of the tests, instantaneous (single-epoch) integer ambiguity resolution was attempted using both single-base and network-based modes of processing. Tables 1 and 2 show the statistics of L1 DD ambiguity resolution for SydNET and SIMRSN respectively. In the tables, the first column is the satellite cut-off elevation angles used in the processing. The second column is the number of DD L1 ambiguities which have been initialised during the period of the tests. The other columns indicate the percentile ambiguity resolution statistics (correct, rejected, wrong) for single-base and network-based techniques. As seen in the tables, the network-based technique performs better, i.e. a higher percentage for the correct fix rates and lower percentages for the rejected fix rates and wrong fix rates, compared to the single-base mode. It also can be noted that, the higher the cut-off elevation angle the better the results for both techniques.

Table 1 Statistics of single-epoch ambiguity resolution for the baseline SPWD-VILL in SydNET

cut-off	Case Initialize	Single-Base			Network-Based		
		Correct %	Reject %	Wrong %	Correct %	Reject %	Wrong %
10°	4103	84.5	5.8	9.7	91.5	3.0	5.6
15°	3916	87.8	2.9	9.3	94.6	1.4	4.0
20°	3345	93.6	0.5	5.9	98.1	0.4	1.5

Table 2 Statistic of single-epoch ambiguity resolution for the baseline LOYA-NYPC in SIMRSN

cut-off	Case Initialize	Single-Base			Network-Based		
		Correct %	Reject %	Wrong %	Correct %	Reject %	Wrong %
10°	4665	96.4	2.1	1.5	98.7	0.8	0.5
15°	3584	97.4	2.4	0.2	99.3	0.7	0
20°	3033	98.5	1.4	0.2	99.6	0.4	0

Figures 7 and 8 highlight the F-ratio validation values for both tests. The figures show that the network-based technique, in most cases, results in higher ratio values than the single-base mode. For this ratio test the critical threshold value is set to 3.

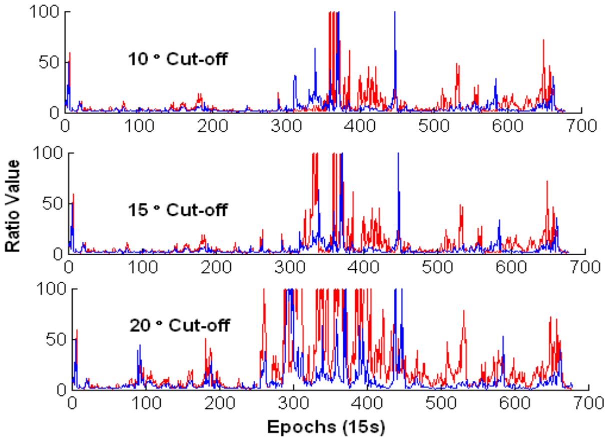


Fig. 7 F-Ratio values of single-base (blue line) and network-based (red line) techniques using various elevation cut-off angles in SydNET (SPWD-VILL)

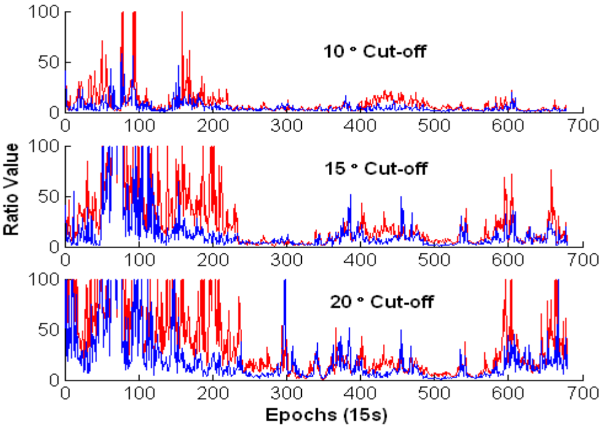


Fig. 8 F-Ratio values of single-base (blue line) and network-based (red line) techniques using various elevation cut-off angles in SIMRSN (LOYA-NYPC)

Further analysis is possible by checking the critical ratio value against the correct and wrong ambiguity results given in Tables 1 and 2. The analysis provides percentages for the ambiguities passed and were correctly accepted, passed but incorrectly rejected (type I error), failed and correctly rejected, failed but incorrectly accepted (type II error), as given in Tables 3 and 4 for SydNET and SIMRSN respectively. It is noted that the results of the network-based technique in both tables give higher percentages for correctly accepted ambiguity using the critical value, and lower percentages in making a type I error, compared to the single-base results. The same conclusion can be made for the correctly rejected wrong ambiguity and the type II error, except in the case of SydNET. Inspecting Table 1, this is only from the percentage calculation. It should be mentioned that the results differ only by applying the network correction or not. Hence, the network correction evidently strengthens the ambiguity resolution and the validation test.

Table 3 Statistics of ambiguity validation for SydNET

	Single-Based				Network-Based			
	Passed %		Failed %		Passed %		Failed %	
cut-off	Acc	Rjct	Acc	Rjct	Acc	Rjct	Acc	Rjct
10°	47.8	52.2	18.3	81.7	58.7	41.3	29.4	70.6
15°	47.5	52.5	19.4	80.6	61.7	38.3	28.5	71.5
20°	66.6	33.4	13.9	86.1	85.1	14.9	20.0	80.0

Table 4 Statistics of ambiguity validation for SIMRSN

	Single-Based				Network-Based			
	Passed %		Failed %		Passed %		Failed %	
Cut-off	Acc	Rjct	Acc	Rjct	Acc	Rjct	Acc	Rjct
10°	55.6	44.4	5.0	95.0	74.6	25.4	4.5	95.5
15°	82.1	17.9	0	100	90.6	9.4	0	100
20°	90.3	9.7	0	100	96.6	3.4	Nil	Nil

After removing the ambiguity biases, the DD L1 measurements are still contaminated by residual distance-dependent errors and station-dependent errors. These biases, together with geometry of the satellites, impact on the positioning results. Based on the fact that the user is static and is a part of the network stations, station-dependent errors such as multipath are assumed to be at a minimum level. During these tests, the geometry of the satellites for both stations was good, with geometric dilutions of precision (GDOP) less than 5 (see Figure 9).

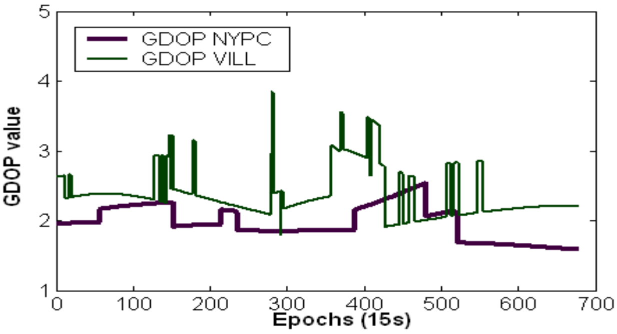


Fig. 9 GDOP values for VILL (SydNET) and NYPC (SIMRSN) during the tests

To reduce distance-dependent errors remaining in the measurements after the removal of the ambiguity biases, dispersive and non-dispersive corrections are applied. Figures 10 and 11 show the DD L1 residuals (for 10º cut-off elevation only) with and without applying the corrections for SydNET and SIMRSN respectively. It can be seen that the network corrections have reduced the magnitude of the residuals compared with the results without the corrections.

Figures 12 and 13 show the results of single-epoch positioning (with and without corrections) after differencing the known positions for VILL and NYPC respectively (for 10º cut-off elevation only). Their corresponding statistics are given in Tables 5 and 6 for each cut-off elevation on both stations. It can be observed from Figures 12 and 13 that the differences in Easting and Northing are at the centimetre level, while the height differences reach the decimetre level, mostly due to residual tropospheric biases.

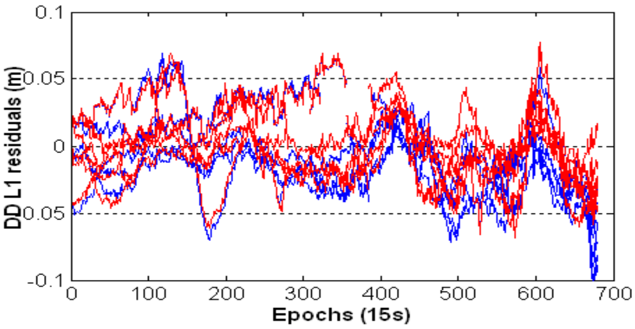


Fig. 10 DD L1 residuals for SPWD-VILL (SydNET), red is with correction and blue is without correction.

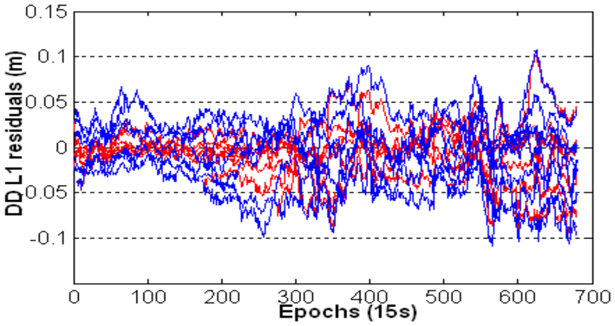


Fig. 11 DD L1 residuals for LOYA-NYPC (SIMRSN), red is with correction and blue is without correction

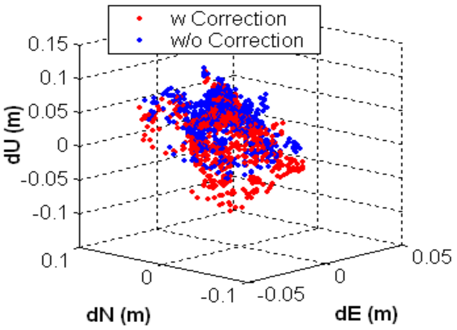


Fig. 12 Differences of calculated L1 positions compared to the known position VILL (SYDNET), red is position calculated with (w) correction and blue is without (w/o) correction

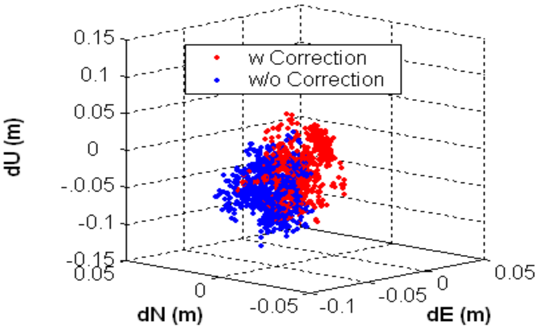


Fig. 13 Differences of calculated L1 positions compared to the known position of NYPC (SIMRSN), red is position calculated with correction and blue is without correction

From Tables 5 and 6 an improvement on the mean Up component (see column 5 of both tables) can be obtained once the corrections are applied. This result can be derived from the non-dispersive correction that reduces the residual tropospheric biases in the measurements. There are no significant differences found in the horizontal components in the case of VILL, but some improvements to the Easting component of NYPC is noticed. It is not clear why the large mean value on Easting component of NYPC were obtained. Perhaps it is because the known position is offset by the 'true' coordinate. Both examples do not indicate much deviation of the coordinate differences in Easting and Northing, however large a variation is noticed in the Up component despite applying the corrections (improvement up to 2.7cm in the case of NYPC at 20° cut-off elevation). In the case of station VILL (SydNET), the Up component variation increases slightly after the correction, but overall, the pattern is reasonable. It shows that applying the correction does not always guarantee better precision of the positioning results, especially in the Up component. It is dependent on the quality of the network corrections and other residual biases that still exist when performing the position computation.

Table 5 Position statistics for VILL (SydNET) with and without (w/o) corrections compared to known position

Cut-off	Corr	Mean (cm)			Deviation (cm)		
		dE	DN	dUp	dE	dN	dUp
10°	w/o	-1.5	-0.6	4.5	1.0	2.5	2.7
	With	-1.0	-0.2	1.8	1.2	2.7	3.9
15°	w/o	-1.5	-0.6	4.4	1.0	2.5	2.8
	With	-1.0	-0.1	1.3	1.1	2.8	3.8
20°	w/o	-1.2	-0.8	2.9	1.3	3.5	3.4
	With	-0.6	-0.6	-0.8	1.3	3.7	4.2

Table 6 Position statistics for NYPC (SIMRSN) with and without (w/o) corrections compared to known position

Cut-off	Corr	Mean (cm)			Deviation (cm)		
		dE	DN	dUp	dE	dN	dUp
10°	w/o	-4.7	0.5	-5.1	1.0	1.0	2.8
	With	-2.4	0.4	-2.8	1.3	0.7	2.8
15°	w/o	-4.5	0.4	-4.4	1.5	1.1	3.5
	With	-2.1	0.5	-1.8	1.8	0.8	2.5
20°	w/o	-4.1	0.4	-5.4	1.5	1.5	5.9
	With	-1.8	0.5	-1.8	1.7	0.9	3.2

5. Concluding Remarks

The ability to capture and model small-scale distance-dependent errors by the network GPS technique enables RTK ambiguity resolution even for longer inter-receiver distances. Information about these distance-dependent errors is included in the network corrections which can be separated into dispersive and non-dispersive components. This separation is useful for advancing network error modelling, and in order to provide more options for the network users' processing strategy.

The dispersive effect that changes rapidly in time and space is modelled as frequently as possible. On the other hand, the slowly and smoothly varying non-dispersive effect is modelled less frequently than the dispersive effect. Furthermore, a running average is applied in order to smooth the non-dispersive correction. For the network user's data processing, this study shows that the separation can be used to improve the IF measurements as well. Such improvement is important especially for real-time ambiguity resolution. The combination of dispersive and non-dispersive corrections is also useful for the user-side computation, if the high quality of both corrections can be assured.

Experiments with local GPS networks in two different geographical locations have demonstrated some advantages of the proposed strategy. Test results and analyses have shown that the proposed strategy performed reasonably well in generating the network correction, resolving the network ambiguities and computing the user's position.

Since both cases show similar ambiguity levels their results can be directly compared. It must be noted that this processing strategy is only available if the user is provided with the measurements of the reference/master station and is able to recognise the network correction components.

References

Chen HY, Rizos C, Han S (2004). ***An instantaneous ambiguity resolution procedure suitable for medium-scale GPS reference station networks***, Survey Review, 37(291), 396-410.

Dai L (2002) ***Augmentation of GPS with GLONASS and pseudolite signals for carrier phase-based kinematic positioning***, PhD Thesis, School of Surveying and Spatial Information Systems, The University of New South Wales, Sydney, Australia, 47-76, 79-107.

Euler HJ, Seeger S, Takac F (2004) ***Analysis of biases influencing successful rover positioning with GNSS-Network RTK***, The 2004 Int. Symp.on GNSS/GPS, Sydney, Australia, 6–8 December, CD-Rom proc.

Hu G, Abbey DA, Castleden N, Featherstone WE, Earls C, Ovstedal O, Weihing D (2005) ***An approach for instantaneous ambiguity resolution for medium to long-range multiple reference station networks***, GPS Solution, 9(1), 1-11.

Han S (1997) ***Carrier phase-based long-range GPS kinematic positioning***, PhD Thesis, School of Surveying and Spatial Information Systems, The University of New South Wales, Sydney, Australia, 46-71.

Hernandes PM, Juan JM, Colombo OL (1999) ***Precise ionospheric determination and its application to real-time GPS ambiguity resolution***, 12th Int. Tech. Meeting of the Satellite Division of the ION, Nashville, Tennessee, 14-17 September, 1409-1417.

Keenan CR, Zebhauser BE, Euler HJ, Wübbena G (2002) ***Using the information from reference station networks: A novel approach conforming to RTCM V2.3 and future V3.0***, IEEE PLANS 2002, Palm Springs, California, 15-18 April, 320-327.

Lynn W, Anil T (1995) ***DGPS architecture based on separating error components, virtual reference stations and FM sub-carrier broadcast***, 51st ION Annual Meeting, 50 Years of Navigation Progress from Art to Utility, Colorado Springs, Colorado, 128-139.

Odijk D (2002) ***Fast Precise GPS Positioning in the Presence of Ionospheric Delays,*** PhD Thesis, Delft University of Technology, Delft, The Netherlands, 69-102.

Rizos C (1997) ***Principles and practice of GPS surveying***, Monograph 17, School of Geomatic Engineering, The University of New South Wales, Sydney, Australia, 288-297.

Sun H, Melgard T, Cannon ME (1999) ***Real-time GPS reference network carrier phase ambiguity resolution***, ION Nat. Tech. Meeting, San Diego, California, 25-27 January, 193-199.

Tajul AM, Samsung L, Rizos C (2005) ***Low latitude troposphere: A preliminary study using GPS CORS data in South East Asia***, ION Nat. Tech. Meeting, San Diego, California, 24-26 January, 685-693.

Wübbena G, Bagge A (1998) ***GNSS multi-station adjustment for permanent deformation analysis networks***, Symp. on Geodesy for Geotechnical & Structural Engineering of the IAG Special Commission 4, Eisenstadt, Austria, 20-22 April, 139-144.

Wanninger L (1997) ***Real-time differential GPS error modeling in regional reference station networks***, IAG Symp 118, Rio de Janeiro, Brazil, 86-92.

Wanninger L (1995) ***Improved ambiguity resolution by regional differential modeling of the ionosphere***, 8th Int. Tech. Meeting of the Satellite Division of the ION, Palm Springs, California, 12-15 September, 55-62.

Architecture and Benefits of an Advanced GNSS Software Receiver

Mark G. Petovello, Cillian O'Driscoll, Gérard Lachapelle, Daniele Borio and Hasan Murtaza
Position, Location And Navigation (PLAN) Group, Department of Geomatics Engineering
University of Calgary

Abstract

This paper describes a GNSS software receiver architecture and the associated benefits in terms of algorithm flexibility and processing efficiency. For the latter, different signal processing algorithms and implementations are considered including processing with a Graphics Processing Unit (GPU); a novel implementation in the GNSS community. The massively parallel processing capability of the GPU is demonstrated relative to other processing optimizations. Sample results of GPS processing are presented including centimetre level positioning. Results obtained with some of the Galileo and GLONASS signals are also included to demonstrate the flexibility of the receiver.

Key words: GNSS, Software Receiver

1. Introduction

Software-based GNSS receivers have been receiving considerable attention in the past several years. Not only do such receivers provide an excellent research tool for investigating and improving GNSS receiver performance in a wide range of conditions, they are also gradually becoming commercially viable, with some companies having already released products to the market (IFEN 2007, Morton 2007, NXP 2007, Scott 2007, CSR 2008, Fastrax 2008). The above advantages are further highlighted by the proliferation of new systems and signals. In contrast, the primary drawback of software receivers is the computational requirements needed to implement the receiver in the first place. In particular, with GNSS sampling rates generally exceeding 4 Msps (samples per second), processing requirements are indeed extreme for the receiver's signal processing operations.

The major objective of a software receiver is therefore to efficiently implement the high rate computations while maintaining the desired flexibility inherent in a software-based approach. Unfortunately, these two objectives are generally at odds with an improvement in one aspect often occurring at the expense of the other. Traditional "hardware-based" GNSS receivers can be viewed as an extreme example of this where the most computationally intense processing is performed using very efficient hardware (i.e., application specific integrated circuits, or ASICs) which is inherently inflexible.

This paper discusses the general design, implementation and testing of a software-based GNSS receiver that addresses the above challenges. The software GSNRx™ (GNSS Software Navigation Receiver)– was developed in C++ and is flexible enough to allow for a wide range of configurations involving different processors, receiver architectures, and acquisition and tracking strategies. With this in mind, the objectives of the paper are two-fold; first, to describe and rationalize the general architecture of the software, and second, to show some sample results obtained with the receiver.

There are two main contributions of the work. First, by presenting the overall software architecture and the underlying motivation for it, it is hoped that readers will gain some insight into the practical implementation issues regarding software receivers. Second, the implementation of a Graphics Processing Unit (GPU) for data processing is presented as a means of improving processing efficiency, even with high sample rates. To the authors' knowledge, this is the first time such an implementation has been used for GNSS software receivers.

The paper begins with a general overview of the software receiver architecture and its corresponding benefits in terms of processing efficiency and algorithm flexibility. The paper discusses how different processors can be incorporated into the receiver and the benefits realized. For example, the receiver can be configured to use a "pure software" approach, or, if available, any other co-processors such as an FPGA (Field Programmable Gate Array) or a GPU (Graphics Processing Unit). The latter is discussed in detail, as this represents a novel implementation for software receivers. It is also demonstrated how the choice of processor can be optimized by making use of any suitable instruction sets available on Intel processors. Following the description of the software, some sample results will be presented that demonstrate the software's capability.

2. GNSS Receiver Methodology

This section describes the basic GNSS receiver methodology, as it applies to the software architecture described in this paper. Algorithm details are available in the cited references. Alternatively, several references on GNSS signal processing in general are available in the public literature including, for example, Van Dierendonck (1995), Misra & Enge (2001), Ma et al (2004), Tsui (2005), Ward et al (2006)and Borre et al (2007). Sample results for more advanced receiver architectures are presented later on, but these architectures are not discussed in detail and the reader is referred to the cited material for more information.

GNSS signal tracking is achieved by generating a local signal within the receiver that matches the incoming signal as closely as possible. This process can be roughly broken down according to Fig. **1** (the acquisition process is roughly similar but some components are omitted for clarity). The different color boxes correspond to the rate at which the operations are performed, as described below. The signal is received at the antenna and is down converted to a lower intermediate frequency (IF) and sampled in the front-end. The front-end and antenna are the only hardware that is strictly necessary in a GNSS receiver. The samples are then passed to any number of individual channels in parallel, each of which is responsible for tracking a given signal that involves Doppler removal and correlation (DRC) also called baseband mixing and de-spreading – tracking error determination and updating of the local signal generator. The remaining steps include the extraction of the navigation data bits (if present on the signal), measurement generation and computation of the navigation solution.

The largest challenge associated with software based GNSS receivers is the computational requirements. To this end, the various operations are divided into high, medium and low rate categories (respectively denoted as red, blue and green boxes in Fig. **1**). In this context, high rate refers to operations performed at the MHz level; typically 4-50 MHz. Medium rate operations are generally performed at a rate of 50-1000 Hz. Low rate operations are generally performed at 20 Hz or less. The various operations are discussed briefly in the following sub-sections.

High Rate Operations

These operations are performed at the sampling rate of the incoming data; typically at 4 Msps or higher. The local signal generation involves computing (i) the sine and cosine of a carrier wave at a particular frequency with a particular starting phase, and (ii) the ranging code starting from a particular code phase. To minimize processing requirements, the sine and cosine of the carrier signals are often generated beforehand and stored in memory for later use (e.g., Ledvina et al 2003, Petovello & Lachapelle 2008). The ranging code may also be computed ahead of time, but is often computed online. It is noted that in most cases, several code phase-shifted versions of the ranging code are required for tracking.

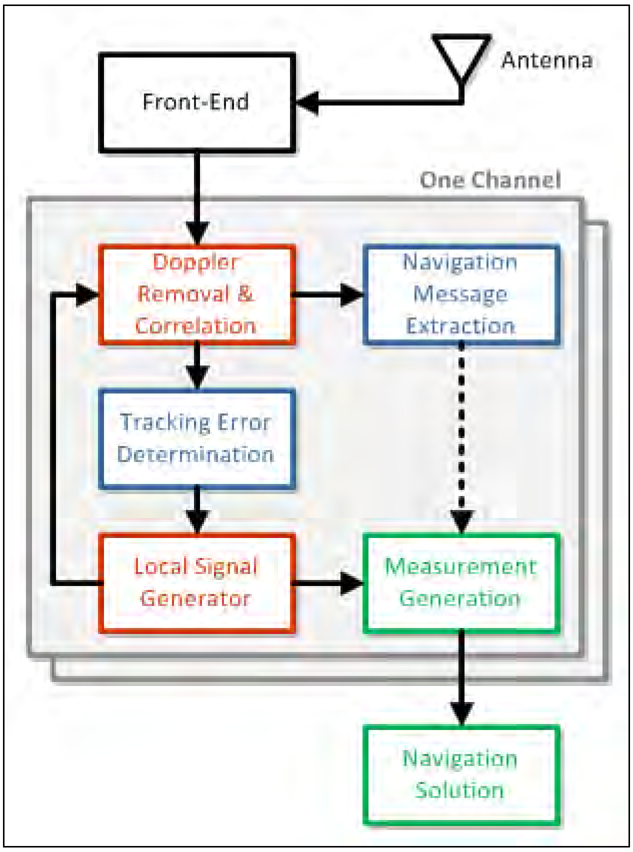


Fig. 1 General overview of GNSS signal tracking (boxes in red, blue and green represent processes performed at high, medium and low rates respectively)

The DRC operation requires projecting the incoming signal on to the locally generated carrier and then correlating the result with the local ranging code. Overall, this requires six multiplications and four additions per sample, per satellite, per code phase (Petovello & Lachapelle 2008). Frequency domain methods (e.g., van Nee & Coenen 1991) are also commonly used for the DRC operation; although mostly used for acquisition, they are also sometimes used for signal tracking as well (Tsui 2005).

The high rate operations, by far, represent the largest computational burden on the receiver. The software architecture should therefore allow for many different processing options in this regard, as will be discussed below.

Medium Rate Operations

The medium rate operations are well understood algorithms and are performed at rates of about 50 Hz to 1 kHz. These operations can be summarized as follows

- Tracking error determination uses a discriminator and loop filter pair to first measure the error (offset) between the incoming and local signals and then filter the result to minimize noise (e.g., Ward et al 2006). Kalman filter-based algorithms may also be implemented here (e.g., Psiaki & Jung 2002, Ziedan & Garrison 2004, Petovello et al 2008a).

- Navigation message extraction is performed only with those signals that broadcast navigation data. The entire operation can be further broken down into bit synchronization, navigation message (frame) synchronization and finally data extraction. For signals that contain a secondary code instead of a navigation message (e.g., the new GPS L5 signal), synchronization with the secondary code is akin to the bit synchronization process.

Low Rate Operations

The low rate operations involve generation of the carrier phase, carrier Doppler and pseudorange measurements and the subsequent computation of the navigation solution. Although some receivers output measurements at 100 Hz, typical rates for mass-market receivers are closer to 1 Hz.

Except in the case of vector-based tracking or ultra-integration with inertial measurement units (IMUs) (e.g., Petovello et al 2008a), the low rate operations are performed independent of the high and medium rate operations described above.

3. Software Architecture

The GSNRx™ software was developed in C++ using a highly modular object-oriented approach. The software was originally written to acquire and track GPS L1 C/A code signals but has since been modified to track many other signals and to use more advanced receiver architectures (more details in the results section). Because of its class-based structure, the architecture will herein be described in terms of "objects" (i.e., instantiated classes).

General Structure

The general architecture adopted for the GSNRx™ software receiver is shown in Fig. **2**. As with Fig. **1**, the boxes refer to the rate at which the operation is performed. Although Fig. **2** is a bit of an abstraction, the basic concept holds true. Before describing the objects in more detail and discussing how they interact, a few things are worth pointing out. First, the term "programmer" is used instead of "user". This is intentional because the purpose of the software is to allow for flexibility in developing and testing various signal tracking algorithms and receiver architectures. That said, the user could effectively be given the same control as the programmer via an appropriate user interface.

The second thing to notice is that in many cases the programmer has control over what objects are created and/or how objects are created. In this way, the software favors an object composition approach. In other words, all classes that adhere to a well defined interface can be used interchangeably allowing the programmer to "create" a particular receiver implementation by simply instantiating the objects with the desired functionality (at compile time or run time, whichever is preferred).

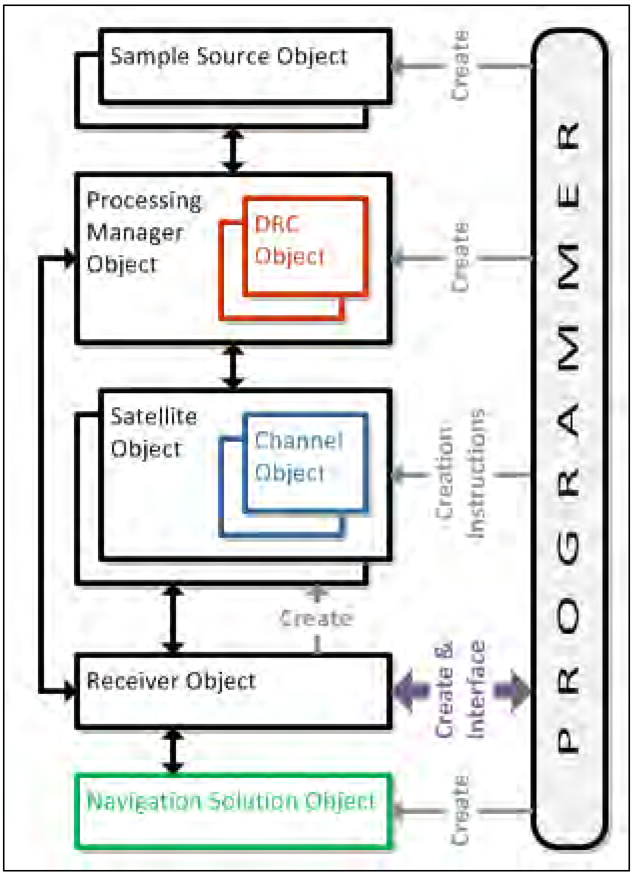


Fig. 2 General Software Architecture of GSNRx™ (colors are used to represent the rate at which each operation is performed, as per Fig. 1)

The third point of interest is that once the necessary objects are created, the programmer only interfaces with a single object— the receiver object. Not only does this improve the readability of the code, but it also simplifies the debugging process because side-effects are avoided.

The final point is that the high, medium and low rate operations are now completely separate. The importance of this will become evident as more details of the implementation are described, as below.

Object Descriptions

The main objects in Fig. **2** are described briefly below. The following section then describes how the objects interact.

Sample Source: A general repository of IF data samples within a given frequency band. The samples can be either real or complex and may be obtained from any practical source (e.g., read from file in post-mission or loaded directly from an analog to digital converter in real-time).

Signal Object: Although not shown in Fig. **2**, a signal is described by its carrier frequency and a ranging code. An example of a signal would be the GPS L1 C/A code, the Galileo E1b code or Galileo E1c code.

Channel Object: A channel is an object that is solely responsible for tracking one or more signals. The flexibility of the channel to handle a wide range of signal combinations (with some limitations) is a major advantage because it allows for more sophisticated tracking algorithms such as data/pilot combining (e.g., Mongrédien et al 2006, Muthuraman et al 2007, Muthuraman et al 2008) or multi-frequency tracking (e.g., Gernot et al 2008a, Gernot et al 2008b). The inputs into the channel are the correlator outputs from the DRC objects (described below).

Satellite Object: A satellite contains one or more channel objects. Satellite objects are responsible for handling satellite-specific information (e.g., different ephemeris messages from different channels). Satellites are created by the receiver on an as-needed basis.

DRC Object: This is an object that performs the DRC operations for a given signal. The algorithm used for this purpose is not defined (e.g., time-domain or frequency-domain, etc.) so long as the interface specifications are met. To this end, the input to the DRC is the sample source and the corresponding signal information from the channels. The outputs are the desired correlator values (more details below).

Processing Manager: The role of the processing manager is to manage the relationships between the channels, signals and sample sources. In so doing, the processing manager has the ability to determine what DRC objects are used for processing. This has major advantages as it allows for highly optimized processing to take place without any modifications to the rest of the code. Several examples of this will be presented later.

Navigation Solution: This object is responsible for computing the position, velocity and time solution (along with any other parameters of interest), typically using least-squares or Kalman filtering estimation algorithms. The navigation solution may also incorporate other sensor information, such as from an IMU, if desired. Having the navigation solution separate from the signal processing components of the receiver (except is some advanced receiver architectures) allows different processing models to be used interchangeably. This is important if an estimation algorithm is better suited to certain applications or operational conditions.

Receiver Object: This is the class that encompasses the entire receiver functionality. As described above, composing the receiver using a variety of objects allows different functionality to be included with only minimal modifications. As stated previously, certain combinations of objects could potentially be selected by the user using an appropriate user interface. The receiver performs all of the necessary high-level operations such as determining what satellites should be acquired and tracked, maintaining the receiver time and interfacing with the user. To this end, the programmer can instruct the receiver how new satellite objects should be created. The receiver object is also responsible for the implementation of vector-based and ultra-tight receiver architectures (e.g., Petovello et al 2008a). In this case, the navigation solution is used to drive the local signal generator directly; the difference between the architectures being that that ultra-tight receiver incorporates an IMU in the navigation solution (see details of navigation solution object) whereas the vector-based receiver does not.

Object Interaction

The general flow of the software is to first create the necessary objects and compose the receiver. The receiver object is informed of what sample sources are available and is also given access to the processing manager. The receiver then creates (allocates) satellite objects as needed based on assumed satellite visibility. As satellite objects are created, information about their channels (and corresponding signals) are passed to the processing manager, which, as described above, is responsible for maintaining the relationships amongst the sample sources, signals and DRC objects. Similarly, as satellites are removed (e.g., because they fall below the local horizon), they are removed from the receiver and the processing manager is informed accordingly.

As samples become available the receiver is told to process the samples, which it does by using the processing manager. To this end, a more detailed view of the interaction of the processing manager, sample sources, channels and signals is shown in Fig. 3. For clarity, only a single sample source, satellite, channel and signal are shown although in practice there may be multiple of each. The processing manager uses the sample source and signal parameters to perform the DRC computations on each signal within the channel. Once the DRC computations are complete, the processing manager forwards all correlator outputs to the channel for processing (tracking). The channels are responsible for informing the processing manager of how many samples to process at a time.

Also shown in Fig. 3 are correlator requests. Correlator requests are initiated by the channel (which has full knowledge of the tracking status for each signal) and are used to request the necessary correlator outputs (code phase and/or frequency offsets relative to the prompt corrrelator) needed for acquisition or tracking. The processing manager is responsible for satisfying the requests of the channels. An example of when a correlator request would be necessary is if early and late correlator spacing is to be narrowed (to mitigate multipath effects) as the tracking status of a particular signal is improved. The key point however, is that the channel completely determines what is needed for tracking the signals contained within it. This is a highly modular structure that is readily modified to accommodate a very wide range of tracking scenarios.

Advantages of the Proposed Architecture

In addition to the flexibility associated with the "composition" approach used in the software and the "autonomous" nature of the channel objects, the proposed architecture offers one other major advantage in terms of processing efficiency. Specifically, it was found that in order to best optimize DRC processing (the high rate computations) on a general processor, all of the data necessary for the DRC computations should be available simultaneously. If this is possible, several optimization approaches can be considered including

Using processor-specific optimizations such as the single instruction, multiple data (SIMD) instruction set available on x86 processors (Pany et al 2003, Heckler & Garrison 2004, Charkhandeh 2007). This is often termed "vectorizing" the processing. Some processors and compilers do this automatically.

- Implementing a multi-threaded architecture which is particularly well suited to multi-core processors.

- Using co-processors such as a field programmable gate array (FPGA), digital signal processor (DSP) or graphical processing unit (GPU).

To date, all of the above optimizations have been implemented (in terms of co-processors, only a GPU implementation is currently complete). Of particular interest here is the use of a GPU which, to the authors' knowledge, has not been previously applied to GNSS software receivers. A GPU is typically used to do computationally expensive graphics coordinate transformations and color shading of polygons in real-time for video games. More recently, they have started to be used for general scientific simulations with great results. A further benefit is that their price-to-performance ratio is several orders of magnitude better than traditional supercomputers.

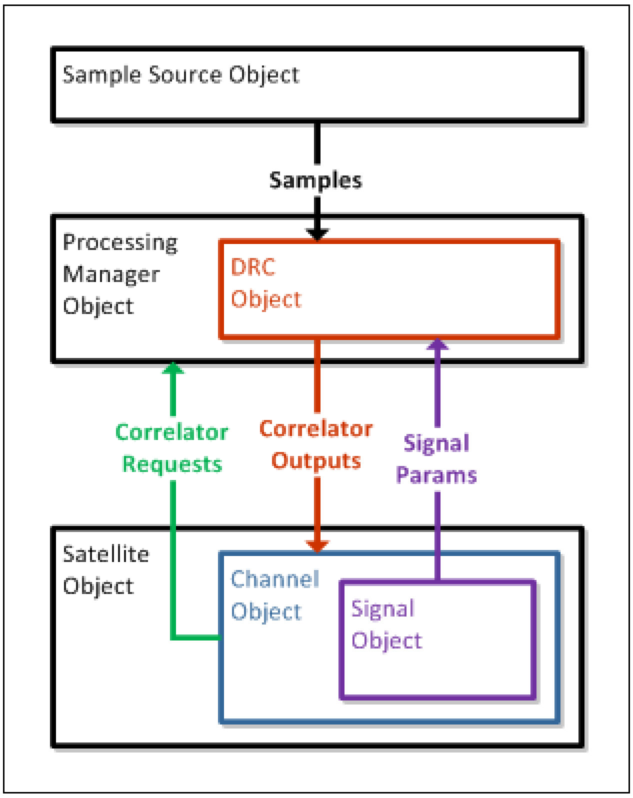


Fig. 3 Interaction of processing manager and associated objects

However, before discussing the GPU implementation in detail (see next section), we first look at different DRC algorithms and their performance using different optimization strategies. In particular, to demonstrate the efficiency of different DRC algorithms, the GSNRx™ software was run using the "rigorous", "table" and "new" DRC algorithms. The "new" algorithm is proposed in Petovello & Lachapelle (2008), which also describes the other two algorithms.

Table **1** shows the average time to perform the DRC processing on 1 ms of data for eight satellites using the different DRCs with different sampling rates. The results were obtained using a single thread on an Intel Xeon quad-core processor. Each processor runs at 1.6 GHz with a 1.066 GHz system bus and 32 kB of L1 cache. Furthermore, each processor has hyper-threading capability, for a total of eight virtual processors.

According to the processor manufacturer, hyper-threading provides "*more efficient use of processor resources*" (Intel 2009), but in practice it has been observed to provide roughly twice the processing capability of a regular processor (i.e., equivalent to using only 50% of the processing core). Comparing the different DRC algorithms, the "table" algorithm performs best. This is somewhat surprising because Petovello & Lachapelle (2008) showed that the "new" algorithm has fewer computations. The difference in performance is explained by the optimizations performed by the compiler and/or processor (different results have been obtained on different processors), and it is clear that these can be significant and should be considered when maximizing processing throughput. Furthermore, for the "table" and "new" algorithms, vectorization provides roughly 50% improvement in processing time.

Table 1 - Average Time to Perform the DRC Processing on 1 ms of Data for Eight Satellites Using Different DRC Algorithms

DRC Algorithm	**Average Time (ms)**	
	5 Msps Data	**25 Msps Data**
Rigorous	1.58	8.06
Rigorous Vectorized	1.29	6.64
Table	1.13	5.54
Table Vectorized	0.56	2.72
New	1.45	6.35
New Vectorized	0.67	3.39

To improve the processing times shown in Table 1, multi-threading was also implemented. In Fig. **4**, the average processing time for 1 ms of data for eight satellites is shown as a function of the number of threads used. The performance of multi-threaded code scales well up to four threads, after which the performance increases are marginal. This is likely due to the fact that, of the eight processors mentioned above, half can be considered "virtual" (i.e., they are not "real" processors). It is expected that these results would scale if more processors were available.

4. DRC Processing Using A GPU

This section presents the details of the GPU implementation for the DRC processing. To this end, two features of GPUs stand out as being particularly useful for the problem at hand. First, they allow a very high degree of parallelism, typically several hundreds or thousands of threads running simultaneously. This is in contrast to the several tens of threads found in a typical central processing unit (CPU). The GPU architecture is limited to executing kernels which perform the same operation on a large data set. While this is sufficient for digital signal processing (as is the case in the current context), it is not at all suitable for implementing general purpose software. The second feature of interest is that GPUs have devoted more silicon area to computationally expensive arithmetic functions. A typical CPU has more than 50% of its area devoted to memory controllers and cache memory. In contrast, a GPU has very little on board cache, devoting extra silicon to arithmetic functions instead. As a result, memory access latency is very high (200-300 clock cycles), but other traditionally expensive operations have been made extremely cheap; on the order of four clock cycles (e.g., sin/cos computations, thread context switching, floating point re-sampling and interpolation).

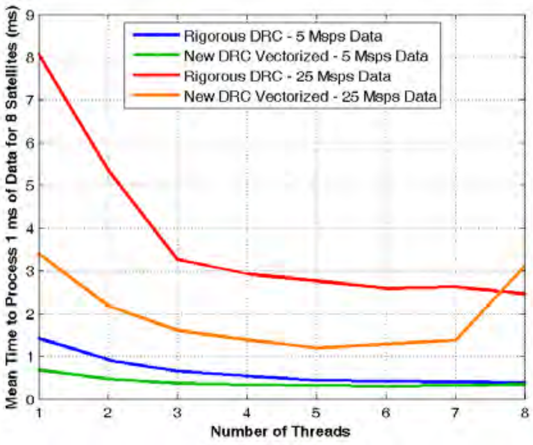


Fig. 4 Mean Time to Perform the DRC Processing on 1 ms of Data for Eight Satellites Using Different DRC Algorithms and Data Rates

5. DRC Processing Using A GPU

This section presents the details of the GPU implementation for the DRC processing. To this end, two features of GPUs stand out as being particularly useful for the problem at hand. First, they allow a very high degree of parallelism, typically several hundreds or thousands of threads running simultaneously. This is in contrast to the several tens of threads found in a typical central processing unit (CPU). The GPU architecture is limited to executing kernels which perform the same operation on a large data set. While this is sufficient for digital signal processing (as is the case in the current context), it is not at all suitable for implementing general purpose software. The second feature of interest is that GPUs have devoted more silicon area to computationally expensive arithmetic functions. A typical CPU has more than 50% of its area devoted to memory controllers and cache memory. In contrast, a GPU has very little on board cache, devoting extra silicon to arithmetic functions instead. As a result, memory access latency is very high (200-300 clock cycles), but other traditionally expensive operations have been made extremely cheap;

on the order of four clock cycles (e.g., sin/cos computations, thread context switching, floating point re-sampling and interpolation).

As mentioned above, relative to a standard CPU, a GPU has considerably more execution units, and each unit is also able to run considerably more threads simultaneously. For example, the NVIDIA 8800GTX GPU used in this work has 16 execution units, each able to run 768 threads simultaneously, for a (theoretical) total of over 12 thousand threads. However, sharing data between multiple processors requires that the processing be partitioned among the different processors and then merged back; a process that can also produce a bottleneck.

In order to better explain how the GPU is used in the software receiver, we recall that to compute any given correlator value, every sample being processed must be multiplied by a local carrier and local code and then the results (one per sample) have to be added up. Obviously, if all of this processing is performed in a single thread on a single processor, there is no benefit to be gained. Instead, within the GPU, the samples are divided into N_s contiguous "slices". Then, for each slice of data, the DRC processing is divided into N_t threads, with each thread processing a subset of the samples within the slice. This concept is shown graphically in Fig. **5**. Both N_s and N_t are design parameters, and for efficiency, should be selected to be powers of two. Each thread then performs the following computations for each sample it is responsible for processing and sums the result:

- Compute of the local code and carrier phase
- Perform the Doppler removal and multiply by the local code

Using the above approach, the GPU effectively divides the processing for a single correlator into a total of $N_s \times N_t$ threads. The result of each thread's processing must then be "reduced", that is, summed to get the final correlator value. However, given the vast number of thread processors on a GPU, this can be highly inefficient and may result in a bottleneck. Fortunately, the NVIDIA architecture provides a mechanism for guaranteeing that a group of threads all have access to a fast "shared" memory. This group of threads is referred to as a "block". Within a block, the reduction process (i.e., adding the results of all the threads) is very efficient because of the shared memory. In GSNRx™, each block is responsible for processing one slice of data for a single correlator, and has been optimized using the techniques of sequential addressing and loop unrolling described by Harris (2007). Finally, because each block only processes a single slice of data, a second reduction is required to obtain the final correlator value by summing the results of all N_s blocks/slices.

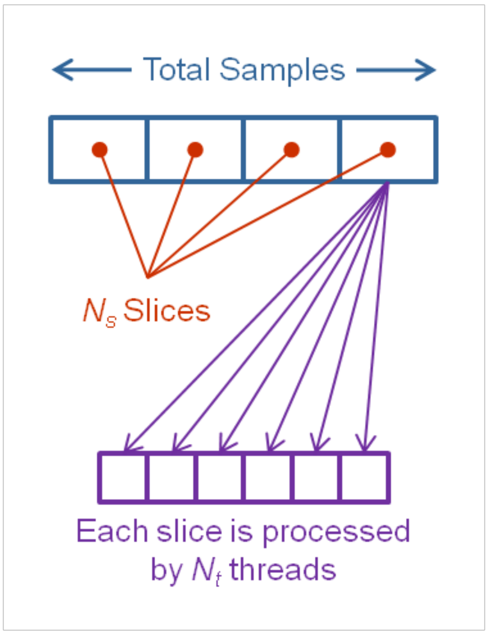


Fig. 5 Diagram Showing How a Group of Samples is Processed in the GPU

The above description only applies to a single correlator value. However, this can be easily extended to include multiple correlator values at once. In this case, processing is performed across a two-dimensional grid of blocks, where the first dimension is the number of correlator outputs and the second dimension is the number of slices (as discussed above). Conceptually, this is shown in Fig. **6**. From the figure, it should be clear that the GPU processing paradigm is highly flexible and scales easily with the number of correlators required and the number of slices the data is divided into (but not the number of samples). In fact, once these parameters are determined (e.g., based on tracking algorithms employed, number of satellites in view, etc.) the GPU takes care of dividing the processing in the most efficient manner possible. In other words, the programmer is allowed to determine the inputs and desired outputs, and then, by using the *same kernel function*, the GPU handles the core processing steps in a transparent manner.

Prior to executing the above processing, all of the samples have to be loaded onto the GPU along with the ranging codes, and the tracking parameters for each correlator to be computed. The transfer of data to the GPU can be executed asynchronously, meaning that the CPU can continue to operate as the samples are loaded onto the GPU. Once this is complete, the execution of the processing is initiated by specifying the kernel function to run, in addition to the grid of blocks and slices discussed above and shown in Fig. **6**.

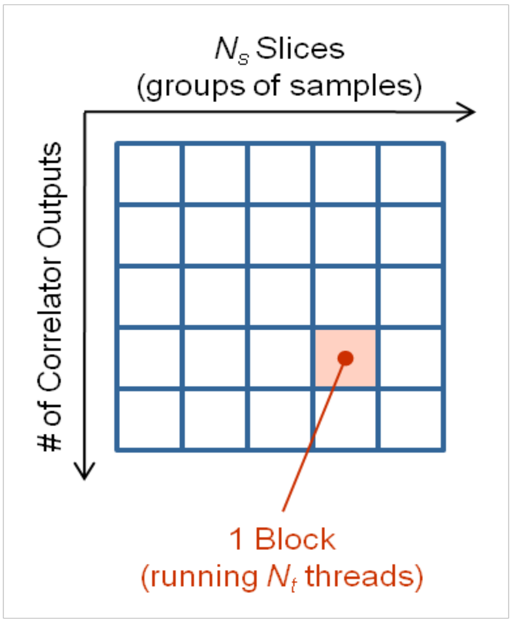


Fig. 6 Computation Grid for GPU Processing

Two final points regarding the GPU implementation are in order before presenting some results. First, GPU programming philosophy is quite distinct from that of CPU programming. In particular, the local code and carrier values are calculated independently for each sample, rather than by incrementing previous values. Second, these local replica values are calculated "on-the-fly" rather than being pre-computed and stored. Specifically, for the local carrier, sin/cos function calls are made explicitly. For a standard CPU, this approach is computationally inefficient, however, due to its design, the GPU executes these function calls much more efficiently. For the local code replica, the code phase (i.e., index into the ranging code that is uploaded to the GPU) is computed independently for each sample using the initial code phase (i.e., at the beginning of the samples to be processed), the code Doppler and sample period. Although this requires more computations than using a lookup table, for example, this is computationally feasible because of the highly parallel structure of the GPU.

To demonstrate the benefit of the GPU, Fig. 7 shows the average DRC processing time for 1 ms of data on eight satellites as a function of the number of threads and slices using 25 Msps data. As can be seen from the plot, there is a tradeoff between the number of slices and the number of threads with the best performance occurring, in this case, with 16 slices and 64 threads per block (although other combinations provide nearly the same performance). Of greater interest however, is that most combinations can process the data in less than 1 ms, suggesting that real-time capability is possible. In contrast, with reference to Fig. **4** (note the different y-axis scales), none of the DRC algorithms were able to process the 25 Msps data in real-time on a general CPU. In other words, the GPU offers the possibility of processing higher data rates in less time, and thus realizing the benefits of the increased signal bandwidths, and at the same time doing it more quickly than with a CPU.

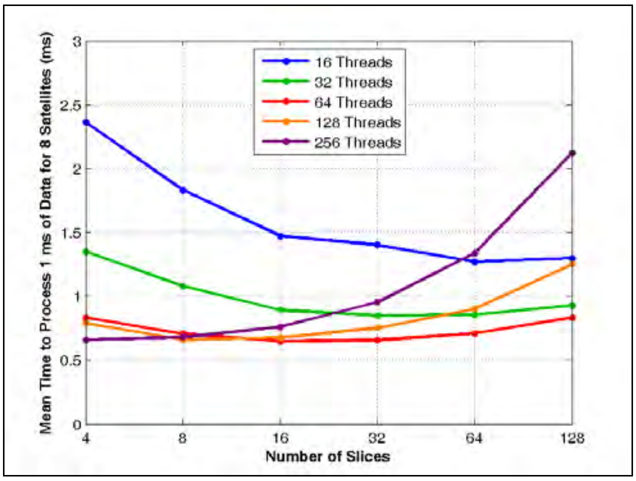


Fig. 7 Mean Time to Perform the DRC Processing on 1 ms of 25 MHz Data for Eight Satellites Using a GPU with Different Numbers of Threads and Slices

6. Software Status and Sample Results

The GSNRx™ software is currently able to acquire and track several signals, as summarized in Table 2. A GPS L1 and Galileo E1 receiver is also working. To date, the navigation solution is only enabled for the GPS L1 signal and the two GLONASS signals because these are the only signals available on a sufficient number of satellites. However, the capability to compute a solution using the other signals is ready and requires final testing with live satellites. In addition, Kalman filter-based, vector-based and ultra-tight architectures (e.g., Petovello et al 2008a) are available for GPS L1 and work is ongoing to incorporate the other signals as well.

Table 2 - Current Status of GSNRx™ Software

Signal	**Status within GSNRx™**
GPS Signals	
L1	Acquire, Track and Navigation Solution
L1C	Work is ongoing
L2C	Acquire and Track
L5	Acquire and Track
Galileo Signals	
E1b/c	Acquire and Track
E5a	Acquire and Track
E5b	Acquire and Track
GLONASS Signals	
L1	Acquire, Track and Navigation Solution
L2	Acquire, Track and Navigation Solution

The software was designed to be able to interface with samples from any front-end hardware, but has not, as yet, been tested in a real-time configuration for any specific front-end. However, given that the results of the previous section show that real-time processing is possible, work is ongoing to have the software work in real-time with a commercially available L1 front-end.

An exhaustive list of references related to the GSNRx™ software (and GNSS signal acquisition and tracking algorithms in general) is beyond the scope of this paper. Interested readers are referred to the PLAN Group website (http://plan.geomatics.ucalgary.ca), which provides access to papers and theses involving software receiver development and testing (as well as all other research topics).

The following sub-sections present some sample results to demonstrate the flexibility of the GSNRx™ software. All of the data processed was collected using a National Instruments front-end system that allows for collection of data on up to three frequency bands at a time (the actual frequency bands used will be clear in the following discussions) using a selectable bandwidth and sampling rate. That said, similar results would be expected with other front-ends. Finally, the results are included mostly to show the flexibility of the software and are therefore presented with minimal explanation.

GPS Results

The most fundamental assessment of a receiver is the standalone positioning error. To this end, Table 3 shows the L1 C/A Code position error statistics for a 15-minute data set collected in open sky conditions. The position solution is accurate to the metre level, as expected given the low level of ionospheric activity during the test and the relatively benign multipath environment in which the data was collected.

Table 3 - GPS L1 Standalone Position Error Statistics

Direction	Mean (m)	RMS (m)
North	-1.4	2.5
East	1.2	1.7
Vertical	-1.1	3.0

The GSNRx™ software is also able to accurately track the carrier phase of the signal, thus allowing high accuracy carrier phase positioning. To illustrate, Fig. **8** shows the RTK position errors as a function of time for a pedestrian-based DGPS test (described in Petovello et al 2007a). For the portion of data shown, the signals were collected in an open sky environment. It is worth noting that the antenna was experiencing peak-to-peak accelerations of about 10 m/s^2 in each coordinate direction throughout the test. In spite of this relatively large level of acceleration, the position errors are still at the centimetre level, as is typical with RTK systems. The error statistics for the data shown in Fig. **8** are given in Table 4.

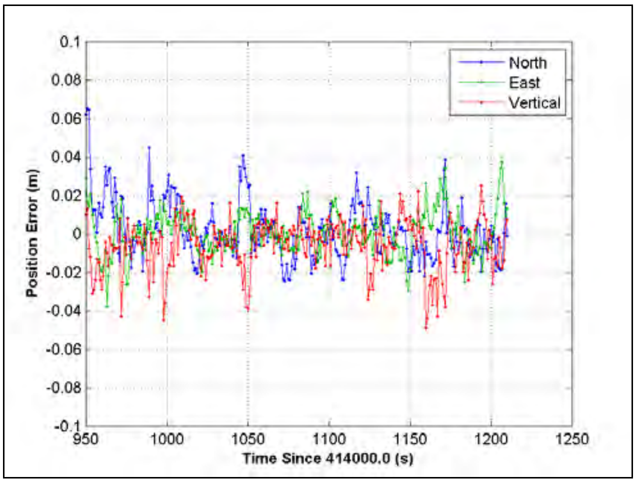


Fig. 8 DGPS L1 RTK Positioning Errors in Open Sky Environment

Table 4 - DGPS L1 RTK Position Error Statistics in Open Sky Environment

Direction	Mean (cm)	RMS (cm)
North	0.2	1.6
East	0.1	1.1
Vertical	-0.6	1.5

In addition to the traditional signal tracking algorithms used to generate the above results, considerable work has also gone into testing new receiver architectures (e.g., Petovello et al 2007a, Petovello et al 2007b, Petovello et al 2008a, Petovello et al 2008b). Two of the most promising architectures are the Kalman filter-based architecture and the ultra-tight integration of GNSS and inertial measurement units (IMUs). A Kalman filter-based receiver replaces the conventional discriminator/loop filter pair with a Kalman filter (although other estimation algorithms could also be used). In an ultra-tight architecture, the IMU measures and compensates for the user's motion, allowing the tracking loops to have a narrower bandwidth. Both the Kalman filter-based and ultra-tight integrations have proven useful when tracking weak GNSS signals. To illustrate this, data was collected on a pedestrian and a variable attenuator was used to slowly reduce the received signal power by 1 dB every 4 s. As the signal power was reduced, different receiver architectures failed at different times. Fig. **9** shows a "histogram" of the horizontal position error as a function of attenuation for different receivers (again, for DGPS L1 RTK positioning). The plot shows the number of epochs whose horizontal position error exceeds a given threshold for all attenuation values up to that shown on the x-axis. Initially, all solutions are able to provide highly accurate solutions, so the number of epochs where the position error exceeds the thresholds is zero. Then, at some point in time (level of attenuation) the receiver "fails" such that corresponding position error exceeds the specified thresholds and never recovers. In this context, "failure" consists of a cycle slip at best, or complete lock of loss at worst. When this happens, the number of epochs where the position exceeds a given threshold increases linearly (with a few minor exceptions). With this in mind, for epochs with an attenuation of 20 dB or less, the standard receiver has about 24 epochs where the horizontal error exceeds 0.1 m. In contrast, for the same level of attenuation, the Kalman filter-based and ultra-tight architectures have one and zero epochs respectively where the horizontal error exceeds 0.1 m. The reason for the improvement with the ultra-tight approach is because in an ultra-tight architecture, the inertial data is used to compensate for receiver motion, thus improving the tracking capabilities of the receiver (*ibid.*).

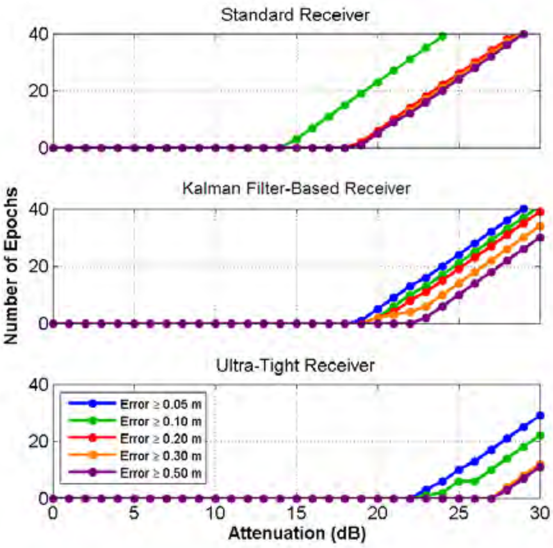


Fig. 9 Horizontal Position Error Histogram for Standard, Kalman Filter-Based and Ultra-Tight Receiver Architectures during Signal Attenuation (lines are plotted in order of increasing position error and thus lines for larger errors may hide those for smaller errors)

New GNSS Signals

As mentioned above, only the GPS L1 signal is fully deployed. The other signals in Table 2 are still not fully available and a fully operational receiver for these signals is not yet feasible (except for the GLONASS signals). Nevertheless, GSNRx™ offers the opportunity to develop, implement and test the acquisition and tracking algorithms for these new signals prior to their full deployment. In so doing, once the signals are available on orbit, the software receiver can be easily extended to take full advantages of these signals, thus reducing product lead time. Included below are some sample results from some ongoing testing and development associated with new GNSS signals and/or systems.

To begin, Fig. **10** shows the acquisition plot for the GIOVE-A (Galileo test satellite) E1b signal employing a BOC(1,1) ranging code. The characteristic side peaks of the signal are clearly visible on each side of the main peak. Also, the sin(x)/x shape is visible in the frequency domain. The results were obtained using two 4-ms coherent integration intervals which are then added non-coherently. Following acquisition, the signal is also able to be tracked (results not shown due to space limitations).

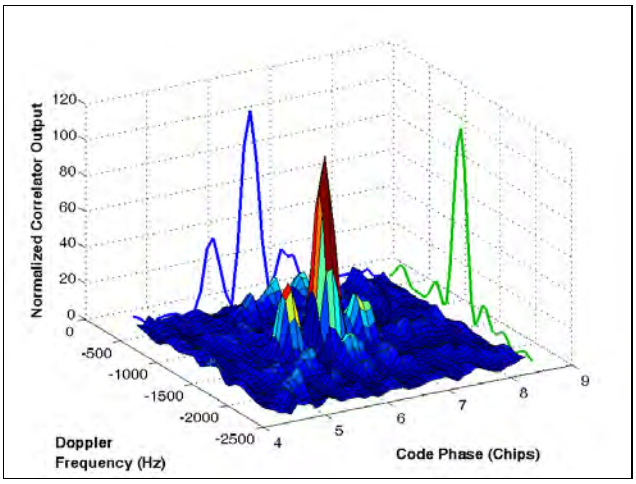


Fig. 10 Acquisition Plot for GIOVE-A E1b Signal with BOC(1,1) Ranging Code (blue line is the projection of the peak in the code phase domain and the green line is the projection of the peak in the frequency domain)

The GSNRx™ software has also been used to acquire and track the GIOVE-A E5b signal. The E5b signal was selected because the Calgary International Airport's distance measuring equipment (DME) falls in this band and it was desired to see if the resulting interference could be effectively mitigated within the receiver. To this end, the upper plot in Fig. **11** shows the power spectral density (PSD) of the original signal as well as the PSD after applying a notch filter inside the receiver. The effect of the DME interference is effectively eliminated by the notch filter. The lower plot in Fig. **11** shows the estimated C/N$_0$ for the two signals and it is obvious that the notch filter allows for better signal tracking. The average improvement in C/N$_0$ is about 2 dB, which is significant.

As a final example, the GSNRx™ software has been used to track signals from the Russian GLONASS system. More specifically, algorithms have been developed to track the civilian signal on both L1 and L2 (Abbasian Nik & Petovello 2008). Table 5 shows the position error statistics for the L1 and L2 position solutions. No GPS measurements were included in these results. For the L1 solution, the position error is very similar to the solution obtained using data collected from a NovAtel OEMV2 receiver (using the same five satellites in both solutions). For the L2-only solution, only four L2-capable satellites were available and the position errors are larger because of satellite geometry degradation, but are still of reasonable magnitude and compare favorably with those of the L1-only solution computed using the same satellites (to remove the effect of satellite geometry).

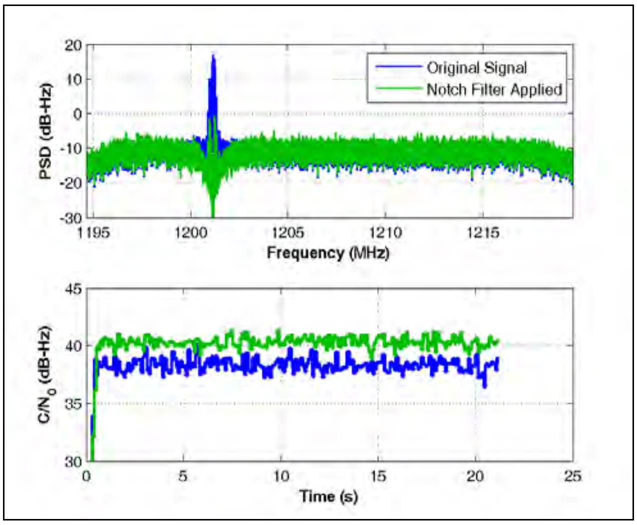


Fig. 11 Power Spectral Density and Estimated C/N$_0$ for GIOVE-A E5b Signal with and without a Notch Filter to Mitigate DME Interference

Table 5 - Standalone GLONASS L1 and L2 Position Error Statistics

Solution	**RMS Error (m)**		
	North	**East**	**Vertical**
5 Satellite Solution			
GSNRx™ (L1-Only)	2.1 m	2.8 m	7.5 m
NovAtel (L1-Only)	0.5 m	1.7 m	2.8 m
4 Satellite Solution			
GSNRx™ (L1-Only)	3.6 m	2.2 m	7.8 m
GSNRx™ (L2-Only)	4.6 m	4.2 m	14.2 m

Summary

From the above, the software architecture clearly provides considerable flexibility to acquire and track new signals and to implement different receiver implementations. This capability is critical in a research environment but is also of interest to agencies wishing to test various algorithms prior to finalizing an implementation in hardware. Another application is products with low replacement rates (e.g., vehicles) that want to incorporate positioning capability now and in the future but would like to easily upgrade functionality in the future as new technologies become available.

7. Summary and Future Work

This paper presented the overall architecture of the GSNRx™ software receiver. The primary benefit of the architecture was shown to be the flexibility it provides for implementing advanced receiver architectures such as ultra-tight integration with an IMU, and for developing and testing algorithms to acquire and test new signals.
In addition, the software is structured to allow processing optimizations to be implemented using whatever resources may be available. Herein, the use of vectorization, multi-threading and a GPU were shown to provide various levels of processing improvements. In particular, the GPU was shown to provide considerable processing improvements, and these are expected to become more significant as more signals need to be tracked simultaneously.

Future work will focus on refining existing algorithms while at the same time incorporating functionality to acquire and track the new signals that will soon be available.

For licensing information, please contact the authors.

REFERENCES

Abbasian Nik, S. and M.G. Petovello (2008) ***Multichannel Dual Frequency GLONASS Software Receiver***, Proceedings of ION GNSS 2008, Savannah, GA, Institute of Navigation, In press.

Borre, K., D. Akos, N. Bertelsen, P. Rinder and S.H. Jenson (2007) ***A Software-Defined GPS and Galileo Receiver,*** A Single-Frequency Approach, Boston, Birkhäuser.

Charkhandeh, S. (2007) ***X86-Based Real Time L1 GPS Software Receiver***, M.Sc. Thesis, Geomatics Engineering, University of Calgary.

CSR (2008) ***CSR eGPS: Fast and reliable positioning - everywhere,*** Retrieved March 5, 2009, from http://www.csr.com/egps/.

Fastrax (2008) ***Smart Positioning with Fastrax Software GPS Receiver***, Fastrax Ltd. 2008.

Gernot, C., K. O'Keefe and G. Lachapelle (2008a) ***Combined L1 / L2C Tracking Scheme for Weak Signal Environment,*** Proceedings of ION GNSS 2008, Savannah, GA, Institute of Navigation, In press.

Gernot, C., K. O'Keefe and G. Lachapelle (2008b) ***Comparison of L1 C/A-L2C Combined Acquisition Techniques,*** Proceedings of European Navigation Conference, Toulouse, France.

Harris, M. (2007) ***Optimizing CUDA, SUPERCOMPUTING 2007 Tutorial***, Retrieved March 5, 2009, from http://www.gpgpu.org/sc2007/.

Heckler, G.W. and J.L. Garrison (2004) ***Architecture of a Reconfigurable Software Receiver***, Proceedings of ION GNSS 2004, Long Beach, CA, Institute of Navigation, 947-955.

IFEN (2007) ***NavX®-NSR - GPS/GALILEO NAVIGATION SOFTWARE RECEIVER***, IFEN, GmbH. 2007, Brochure for NavX®-NSR.

Intel (2009) ***Hyper-Threading Technology***, Retrieved 24 March, 2009, from http://www.intel.com/technology/platform-technology/hyper-threading/index.htm?iid=tech_product+ht.

Ledvina, B.M., S.P. Powell, P.M. Kintner and M.L. Psiaki (2003) ***A 12-Channel Real-Time GPS L1 Software Receiver***, Proceedings of ION National Technical Meeting, Anaheim, CA, Institute of Navigation, 767-782.

Ma, C., G. Lachapelle and M.E. Cannon (2004) ***Implementation of a Software GPS Receiver,*** Proceedings of ION GNSS 2004, Long Beach, CA, Institute of Navigation, 956-970.

Misra, P. and P. Enge (2001) ***Global Positioning System Signals, Measurement, and Performance***, Lincoln, MA, Ganga-Jamuna Press.

Mongrédien, C., G. Lachapelle and M.E. Cannon (2006) ***Testing GPS L5 Acquisition and Tracking Algorithms Using a Hardware Simulator,*** Proceedings of ION GNSS 2006, Fort Worth, TX, Institute of Navigation, 2901-2913.

Morton, J. (2007) ***Expert Advice: Software Defines Future, GPS World System Design and Test News***, Retrieved January 7, 2008, from http://sidt.gpsworld.com/gpssidt/Expert+Advice+%26+Leadership+Talks/Expert-Advice-mdash-Software-Defines-Future/ArticleStandard/Article/detail/445464?contextCategoryId=35358&searchString=software%20receiver.

Muthuraman, K., R. Klukas and G. Lachapelle (2008) ***Performance Evaluation of L2C Data/Pilot Combined Carrier Tracking***, Proceedings of ION GNSS 2008, Savannah, GA, Institute of Navigation, 9 pages.

Muthuraman, K., S.K. Shanmugam and G. Lachapelle (2007) ***Evaluation of Data/Pilot Tracking Algorithms for GPS L2C Signals Using Software Receiver,*** Proceedings of ION GNSS 2007, Fort Worth, TX, Institute of Navigation, 11 pages.

NXP (2007) ***NXP Software teams with Mango Research on high performance Personal Navigation Device,*** Retrieved March 5, 2009, from http://www.software.nxp.com/?pageid=140.

Pany, T., S.W. Moon, M. Irsigler, B. Eissfeller and K. Fürlinger (2003) ***Performance Assessment of an Under Sampling SWC Receiver for Simulated High-Bandwidth GPS/Galileo Signals and Real Signals***, Proceedings of ION GPS/GNSS 2003, Portland, OR, Institute of Navigation, 103-116.

Petovello, M.G. and G. Lachapelle (2008) ***Centimeter-Level Positioning Using an Efficient New Baseband Mixing and De-Spreading Method for Software GNSS Receivers,*** Journal on Advances in Signal Processing (JASP), In Press.

Petovello, M.G., C. O'Driscoll and G. Lachapelle (2007a) ***Ultra-Tight GPS/INS for Carrier Phase Positioning In Weak-Signal Environments***, Proceedings of NATO RTO SET-104 Symposium on Military Capabilities Enabled by Advances in Navigation Sensors, Antalya, Turkey, NATO, 18 pages.

Petovello, M.G., C. O'Driscoll and G. Lachapelle (2008a) ***Carrier Phase Tracking of Weak Signals Using Different Receiver Architectures,*** Proceedings of ION National Technical Meeting, San Diego, CA, Institute of Navigation, 781-791.

Petovello, M.G., C. O'Driscoll and G. Lachapelle (2008b) ***Weak Signal Carrier Tracking Using Extended Coherent Integration with an Ultra-Tight GNSS/IMU Receiver***, Proceedings of European Navigation Conference, Toulouse, France, 11 pages.

Petovello, M.G., K. O'Keefe, G. Lachapelle and M.E. Cannon (2007b) ***Consideration of Time-Correlated Errors in a Kalman Filter Applicable to GNSS***, Journal of Geodesy, In Press.

Psiaki, M.L. and H. Jung (2002) ***Extended Kalman Filter Methods for Tracking Weak GPS Signals,*** Proceedings of ION GPS 2002, Portland, OR, Institute of Navigation, 2539-2553.

Scott, L. (2007) Directions 2008: ***Software-Defined Radio Role to Grow, GPS World System Design and Test News,*** Retrieved January 7, 2008, from http://sidt.gpsworld.com/gpssidt/Receiver+Design/Directions-2008-Software-Defined-Radio-Role-to-Gro/ArticleStandard/Article/detail/476704.

Tsui, J.B.-Y. (2005) ***Fundamentals of Global Positioning System Receivers: A Software Approach,*** Hoboken, NJ, John Wiley & Sons, Inc.

Van Dierendonck, A.J. (1995) ***GPS Receivers, Global Positioning System: Theory and Applications,*** B. W. Parkinson and J. J. Spilker, Jr., American Institute of Aeronautics and Astronautics, Inc. I, 329-407.

van Nee, D.J.R. and A.J.R.M. Coenen (1991) ***New fast GPS code-acquisition technique using FFT,*** Electronics Letters, 27(2), 158-160.

Ward, P.W., J.W. Betz and C.J. Hegarty (2006) ***Satellite Signal Acquisition, Tracking, and Data Demodulation,*** Understanding GPS Principles and Applications, E. D. Kaplan and C. J. Hegarty, Norwood, MA, Artech House, Inc., 153-241.

Ziedan, N.I. and J.L. Garrison (2004) ***Extended Kalman Filter-Based Tracking of Weak GPS Signals under High Dynamic Conditions***, Proceedings of ION GNSS 2004, Long Beach, CA, Institute of Navigation, 20-31.

Technical Notes

"Technical Notes" is a new column in this Journal, featuring reviews of technical or theoretical tools for topics of positioning systems and their applications. Specialists in various fields are welcome to contribute a normal article to outline the issues of interest as systematic possible. In general, the manuscripts may aim to fill the gaps between the textbooks and scientific papers published for the specific topic. In this issue, Dr Jianguo Wang, York University, will review testing statistics in the context of Kalman filter, providing useful tools for design and use of a Kalman filter in navigation and positioning applications.

The column of this issue is coordinated by Dr Jianguo Wang, who appreciates your contribution to this column, along with your comments or ideas for topics for future issues (jgwang@yorku.ca).

GEOSPATIAL SCIENCE AND TECHNOLOGY
MANUAL OF
Edited by John D. Bossler
WIRELESS POSITIONING
TECHNOLOGIES AND APPLICATIONS
MONTENBRUCK GILL
Satellite Orbits
WIRELESS SENSOR NETWORKS
Intelligent Vehicle Technology and Trends
Bishop
KUPPER
LOCATION-BASED SERVICES
FUNDAMENTALS AND OPERATION
WILEY
Global Positioning System: Theory and Applications II
Parkinson and Spilker, Eds. / Axelrad and Enge, Assoc. Eds
AIAA
Global Positioning System: Theory and Applications I
Parkinson and Spilker, Eds. / Axelrad and Enge, Assoc. Eds
AIAA
Van Sickle
GPS for Land Surveyors
VI Hotine-Marussi Symposium on Theoretical and Computational Geodesy
SOMMERVILLE
SOFTWARE ENGINEERING
ADDISON-WESLEY

Calibration and Stochastic Modelling of Inertial Navigation Sensor Errors

Mohammed El-Diasty and Spiros Pagiatakis
Dept. of Earth & Space Science & Engineering, York University, Canada

Abstract

The integration of Global Positioning System (GPS) with an inertial measurement unit (IMU) has been widely used in many applications of positioning and orientation. The performance of a GPS-aided inertial integrated navigation system is mainly characterized by the ability of the IMU to bridge GPS outages. This basically depends on the inertial sensor errors that cause a rapid degradation in the integrated navigation solution during periods of GPS outages. The inertial sensor errors comprise systematic and random components. In general, systematic errors (deterministic) can be estimated by calibration and therefore they can be removed from the raw observations. Random errors can be studied by linear or high order nonlinear stochastic processes. These stochastic models can be utilized by a navigation filter such as, Kalman filter, to provide optimized estimation of navigation parameters. Traditionally, random constant (RC), random walk (RW), Gauss-Markov (GM), and autoregressive (AR) processes have been used to develop the stochastic model in the navigation filters.

In this technical note, the inertial sensor errors are introduced and discussed. Subsequently, a six-position laboratory calibration test is described. Then, mathematical models for RC, RW, GM, and AR stochastic models with associated variances for gyros and accelerometer random errors are presented along with a discussion regarding ongoing research in this field. Also, the implementation of a stochastic model in a loosely coupled INS/GPS navigation filter is explained.

Keywords: GPS, INS, Calibration, Random Error, Stochastic Process.

1. Inertial Sensor Errors

The performance of a GPS-aided inertial navigation system is mainly characterized by the ability of the IMU to bridge GPS outages. This ability of the IMU to bridge GPS outages depends on the inertial sensor errors, which, if not treated properly, cause a rapid degradation in the integrated navigation solution during the periods of GPS outages. Inertial sensors are used to collect measurements that can be processed using inertial processing software to estimate position, velocity, and attitude that can be integrated with GPS data to provide a complete navigation solution. An inertial sensor is made up of three gyroscopes (shortly gyros), and three accelerometers. A gyro is device that maintains orientation in space, and thus can sense the rate of change of direction (angular rate) of the vehicle on which is mounted. The rate of change of direction (angular rate) can mathematically be integrated to provide attitude changes over time. Similarly, an accelerometer senses linear accelerations, which when integrated in time give velocity changes, and when integrated twice give position changes over time. The major error sources in gyros and accelerometers are biases, and scale errors related to non-orthogonalities of the axes. Hence, due to the integration process, biases and scale errors impose unstable errors in positions, velocities, and attitudes. The growth of these errors depends on the type of inertial sensor used (high, medium and low grade). The inertial sensor errors can be classified into two types, deterministic (systematic) and random (Nassar, 2005).

Major deterministic error sources include bias and scale errors, which can be removed by specific calibration procedures; Park and Gao (2002) discussed such laboratory calibration procedures. However, the inertial sensor random errors primarily include the sensor noise, which consists of two parts, a high frequency and a low frequency component (Skaloud et al., 1999). The high frequency component has white noise characteristics, while the low frequency component is characterized by correlated noise (Skaloud et al., 1999). De-noising methodology is required to filter the high frequency noise in the inertial sensor measurements prior to processing, using a low pass filter, a wavelet or neural network de-noising procedure (El-Rabbany and El-Diasty, 2004). Several studies have focused on evaluating such techniques (Skaloud et al., 1999; Nassar, 2005; Abdel-Hamid et al., 2004). On the other hand, the low frequency noise component (correlated noise) can be modelled using random processes such as, random constant, random walk, Gauss-Markov or periodic random processes (Nassar, 2005). The most commonly used process is the first-order Gauss-Markov process. The development of the stochastic model for an inertial sensor is one of the most important steps for building a reliable integrated navigation system. The reason is that the inertial sensor propagates large navigation errors in a small time interval. Unless an accurate stochastic model is developed, the mechanization parameters (velocity, attitude, position) will be contaminated by the unmodelled errors and the system performance will be degraded (El-Diasty et al., 2007b).

Let us assume that inertial sensor measurements are denoted by ω_{imu} and f_{imu} representing direction rate of change (angular rate) and linear acceleration, respectively. They can be written approximately as functions of the true direction rate of change ω and the true linear acceleration f in the body frame (because very small inertial sensor second order errors are neglected) as (Titterton, 2004; El-Diasty et al., 2007b):

$$\omega_{imu} \approx [I + S_g + \delta S_g]\omega + b_g + \delta b_g + w_g\,, \qquad (1)$$

$$f_{imu} \approx [I + S_a + \delta S_a]f + b_a + \delta b_a + w_a\,, \qquad (2)$$

where:

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, S_{g,a} = \begin{bmatrix} s_{XX} & s_{XY} & s_{XZ} \\ s_{YX} & s_{YY} & s_{YZ} \\ s_{ZX} & s_{ZY} & s_{ZZ} \end{bmatrix}_{g,a},$$

$$\delta S_{g,a} = \begin{bmatrix} \delta s_{XX} & \delta s_{XY} & \delta s_{XZ} \\ \delta s_{YX} & \delta s_{YY} & \delta s_{YZ} \\ \delta s_{ZX} & \delta s_{ZY} & \delta s_{ZZ} \end{bmatrix}_{g,a}, b_{g,a} = \begin{bmatrix} b_X \\ b_Y \\ b_Z \end{bmatrix}$$

$$\delta b_{g,a} = \begin{bmatrix} \delta b_X \\ \delta b_Y \\ \delta b_Z \end{bmatrix}_{g,a}, w_{g,a} = \begin{bmatrix} w_X \\ w_Y \\ w_Z \end{bmatrix}_{g,a}$$

where *I* is the identity matrix (unitless), S_g and S_a are matrices (unitless) comprising scale errors (diagonal elements) and non-orthogonality errors (non-diagonal elements) of the gyro and accelerometer respectively, b_g and b_a are biases (deg/s for gyros and m/s^2 for accelerometers), δS_g and δS_a are matrices (unitless) comprising residual scale errors (diagonal elements) and residual non-orthogonality errors (non-diagonal elements), δb_g and δb_a are residual biases (deg/s for gyros and m/s^2 for accelerometers), and w_g and w_a are zero mean white noises (deg/s for gyros and m/s^2 for accelerometers). Biases and scale errors are either estimated through laboratory calibration or can be modelled as additional parameters in Kalman filter. In this study we discuss the laboratory calibration approach that allows the direct estimation of the bias and scale, which we can then remove from the raw measurements ω_{imu} and f_{imu} (i.e. before implementing the inertial mechanization equations). Then, the corrected measurements ω_{ib}^{b} and f^{b} (which will be the input to inertial mechanization equations) are:

$$\omega_{ib}^{b} \approx [I + (\delta S_g)]\omega + (\delta b_g) + w_g\,, \qquad (3)$$

$$f^{b} \approx [I + (\delta S_a)]f + (\delta b_a) + w_a\,. \qquad (4)$$

However, ω_{ib}^{b} and f^{b} still contain random errors: δS_g and δS_a are matrices comprising residual scale errors (diagonal elements) and residual non-orthogonality errors of the gyro and accelerometer respectively, δb_g and δb_a are residual biases, and w_g and w_a are zero mean white noises.

The residual biases and residual scale errors are the inertial random errors and can usually be modelled by stochastic models inside a Kalman filter at each epoch and then removed simultaneously from ω_{ib}^{b} and f^{b} (epoch by epoch) during the mechanization equation implementation. This stochastic model can be random constant, random walk, or Gauss-Markov process (Grewal et al., 2007; El-Diasty et al., 2007b). The resultant measurements at each epoch are $\hat{\omega}_{ib}^{b}$ and $\hat{f}^{b}$, which represent the optimal estimation of the gyro and accelerometer outputs and they can be used to provide an accurate and continuous navigation solution. In the next section we discuss the six-position calibration laboratory test used to estimate the gyro and accelerometer biases and their scale errors (b_g, b_a, S_g, S_a).

2. Laboratory Determinations of Inertial Biases, Scale, and Non-orthogonality Errors

The laboratory calibration of an IMU is well documented in Titterton (2004) and Salychev (1998). Also, Shin and El-Sheimy (2002), and Syed et al. (2007) are two key papers that describe the practical implementation for these calibration methods and show ongoing research in the area of inertial navigation. In laboratory calibration, a six-position static test (up and down position for the three inertial sensor axes) is commonly performed to collect the gyro and accelerometer measurements. From this test, an estimate of the gyro and accelerometer bias, scale, and non-orthogonality errors can be obtained. The bias is a systematic error called bias offset, which is the offset of the sensor measurement from its true value. The scale error describes the error in the relationship between the sensor output signal and the measured physical quantity. The non-orthogonality error is the error resulting from the imperfection of mounting the inertial sensors along three orthogonal axes at the time of manufacturing. Fig. 1 shows the up and down positions and the excitation (reference) signal in each position (we have a natural excitation signal for accelerometers which is local gravity g in the lab (Salychev, 1998)). We excite the gyro by a known rotational rate ω_{known} using a calibration turntable (Titterton, 2004). Therefore, all three accelerometers can be tested using two-position static tests in the zenith direction, and any gyro sensor can be tested using a two-position dynamic test in any direction (Titterton, 2004). It should be noted that the direction of the known rotational rate ω_{known} in Fig. 1 is the clockwise direction for both, up and down positions.

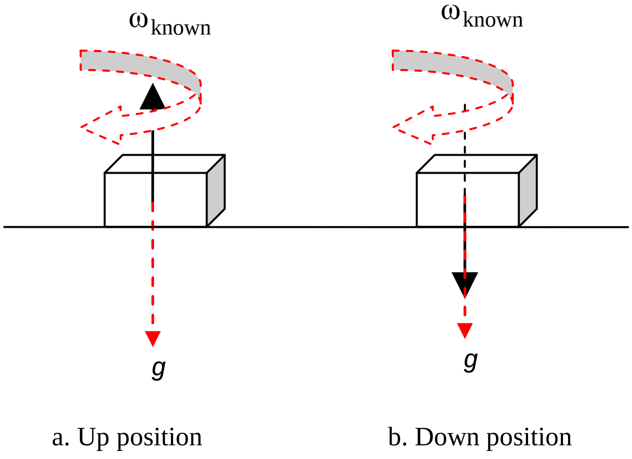


Fig. 1 Up and down positions of the IMU for calibration of one axis – the dotted arrows describe the true excitation (reference) signal.

There are two methodologies that can be employed to find the calibration parameters (b_g, b_a, S_g and S_a). In this technical note, we simply call these two methods six-position direct method and six-position weighted least squares method. In both methods, data are collected in each position for a minimum of 30 seconds (which can give 3000 samples if the sampling rate is 100 Hz). Then the gyro and accelerometer biases (b_g and b_a), scale, and non-orthogonality error matrices (S_g and S_a) can be calculated. It should be noted that S_g and S_a are matrices comprising the scale errors (diagonal elements) and non-orthogonality errors (non-diagonal elements).

2.1 Six-Position Direct Method

Assume that we want to calibrate the X-axis gyro and accelerometer errors of an IMU. In the direct method the biases and scale errors can only be estimated (but non-orthogonality errors are neglected) from the two positions (X-axis up and X-axis down) by taking the average of the measurements in three steps as follows:

Step1: assume that the gyro and accelerometer measurements at epoch *k* are:

$$\omega_{imu}^{Xup}\Big|_k \approx (1+s_{XXg}+\delta s_{XXg})\cdot(-\omega_{\text{known}})+ b_{Xg}+\delta b_{Xg}+w_{Xg} \quad , \qquad (5)$$

$$f_{imu}^{Xup}\Big|_k \approx (1+s_{XXa}+\delta s_{XXa})\cdot(-g)+ b_{Xa}+\delta b_{Xa}+w_{Xa} \quad , \qquad (6)$$

$$\omega_{imu}^{Xdn}\Big|_k \approx (1+s_{XXg}+\delta s_{XXg})\cdot(\omega_{\text{known}})+ b_{Xg}+\delta b_{Xg}+w_{Xg} \quad , \qquad (7)$$

$$f_{imu}^{Xdn}\Big|_k \approx (1+s_{XXa}+\delta s_{XXa})\cdot(g)+ b_{Xa}+\delta b_{Xa}+w_{Xa} \quad , \qquad (8)$$

where Xup means IMU X-axis is in up direction, Xdn means X-axis is in down direction, ω_{known} is known rotational rate and g is the local gravity.

Step2: average the gyro and accelerometer measurements as follows:

$$Av(\omega_{imu}^{Xup}) \approx (1+s_{XXg})\cdot(-\omega_{\text{known}})+b_{Xg}, \qquad (9)$$

$$Av(f_{imu}^{Xup}) \approx (1+s_{XXa})\cdot(-g)+b_{Xa}, \qquad (10)$$

$$Av(\omega_{imu}^{Xdn}) \approx (1+s_{XXg})\cdot(\omega_{\text{known}})+b_{Xg}, \qquad (11)$$

$$Av(f_{imu}^{Xdn}) \approx (1+s_{XXa})\cdot(g)+b_{Xa}, \qquad (12)$$

where Av is the average operator. It should be noted that when the measurements are averaged for one position, noise (w_{Xg} and w_{Xa}) and residual errors ($\delta s_{XXg}, \delta b_{Xg}, \delta s_{XXa}$ and δb_{Xa}) are eliminated because their expected values (mean) are zeros.

Step3: estimate the bias and scale errors using the following equations:

$$b_{Xg} = \frac{Av(\omega_{imu}^{Xdn}) + Av(\omega_{imu}^{Xup})}{2} \ , \qquad (13)$$

$$s_{XXg} = \frac{Av(\omega_{imu}^{Xdn}) - Av(\omega_{imu}^{Xup}) - 2 \cdot \omega_{known}}{2 \cdot \omega_{known}} , \qquad (14)$$

$$b_{Xa} = \frac{Av(f_{imu}^{Xdn}) + Av(f_{imu}^{Xup})}{2} , \qquad (15)$$

$$s_{XXa} = \frac{Av(f_{imu}^{Xdn}) - Av(f_{imu}^{Xup}) - 2 \cdot g}{2 \cdot g} . \qquad (16)$$

The bias and scale errors for Y-axis and Z-axis can be estimated using the same approximate method and steps when Y-axis and Z-axis are configured in the up and down positions, respectively. In this methodology we use two measurements for each axis to estimate the biases and scale errors (in total six measurements are used for the three axes). The advantage of this method lies in its simplicity of implementation. However, the disadvantage is that the non-orthogonality errors can not be estimated.

2.2 Six-Position Weighted Least Squares Method

In this method, all biases, scale errors and non-orthogonality errors for the three axes X, Y and Z are estimated using all the measurements from the six-position configuration. Assume that we wish to estimate the accelerometer errors. From the six-position test we expect to have the following observation equations:

1. when the X-axis is in the up direction, we estimate three averages from the three accelerometers:

$$\begin{bmatrix} Av(f_{imu}^{Xup})_X \\ Av(f_{imu}^{Xup})_Y \\ Av(f_{imu}^{Xup})_Z \end{bmatrix} = \begin{bmatrix} b_X \\ b_Y \\ b_Z \end{bmatrix}_a + \left[\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} s_{XX} & s_{XY} & s_{XZ} \\ s_{YX} & s_{YY} & s_{YZ} \\ s_{ZX} & s_{ZY} & s_{ZZ} \end{bmatrix}_a \right] \cdot \begin{bmatrix} -g \\ 0 \\ 0 \end{bmatrix} \qquad (17)$$

2. when the X-axis is in the down direction we estimate three averages from the three accelerometers:

$$\begin{bmatrix} Av(f_{imu}^{Xdn})_X \\ Av(f_{imu}^{Xdn})_Y \\ Av(f_{imu}^{Xdn})_Z \end{bmatrix} = \begin{bmatrix} b_X \\ b_Y \\ b_Z \end{bmatrix}_a + \left[\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} s_{XX} & s_{XY} & s_{XZ} \\ s_{YX} & s_{YY} & s_{YZ} \\ s_{ZX} & s_{ZY} & s_{ZZ} \end{bmatrix}_a \right] \cdot \begin{bmatrix} g \\ 0 \\ 0 \end{bmatrix} \qquad (18)$$

3. when the Y-axis is in the up direction we estimate three averages from the three accelerometers:

$$\begin{bmatrix} Av(f_{imu}^{Yup})_X \\ Av(f_{imu}^{Yup})_Y \\ Av(f_{imu}^{Yup})_Z \end{bmatrix} = \begin{bmatrix} b_X \\ b_Y \\ b_Z \end{bmatrix}_a + \left[\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} s_{XX} & s_{XY} & s_{XZ} \\ s_{YX} & s_{YY} & s_{YZ} \\ s_{ZX} & s_{ZY} & s_{ZZ} \end{bmatrix}_a \right] \cdot \begin{bmatrix} 0 \\ -g \\ 0 \end{bmatrix} \qquad (19)$$

4. when the Y-axis is in the down direction we estimate three averages from the three accelerometers:

$$\begin{bmatrix} Av(f_{imu}^{Xdn})_X \\ Av(f_{imu}^{Xdn})_Y \\ Av(f_{imu}^{Xdn})_Z \end{bmatrix} = \begin{bmatrix} b_X \\ b_Y \\ b_Z \end{bmatrix}_a + \left[\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} s_{XX} & s_{XY} & s_{XZ} \\ s_{YX} & s_{YY} & s_{YZ} \\ s_{ZX} & s_{ZY} & s_{ZZ} \end{bmatrix}_a \right] \cdot \begin{bmatrix} 0 \\ g \\ 0 \end{bmatrix} \qquad (20)$$

5. when the Z-axis is in the up direction we estimate three averages from the three accelerometers:

$$\begin{bmatrix} Av(f_{imu}^{Zup})_X \\ Av(f_{imu}^{Zup})_Y \\ Av(f_{imu}^{Zup})_Z \end{bmatrix} = \begin{bmatrix} b_X \\ b_Y \\ b_Z \end{bmatrix}_a + \left[\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} s_{XX} & s_{XY} & s_{XZ} \\ s_{YX} & s_{YY} & s_{YZ} \\ s_{ZX} & s_{ZY} & s_{ZZ} \end{bmatrix}_a \right] \cdot \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \qquad (21)$$

6. when the Z-axis is in the down direction we estimate three averages from the three accelerometers:

$$\begin{bmatrix} Av(f_{imu}^{Zdn})_X \\ Av(f_{imu}^{Zdn})_Y \\ Av(f_{imu}^{Zdn})_Z \end{bmatrix} = \begin{bmatrix} b_X \\ b_Y \\ b_Z \end{bmatrix}_a + \left[\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} s_{XX} & s_{XY} & s_{XZ} \\ s_{YX} & s_{YY} & s_{YZ} \\ s_{ZX} & s_{ZY} & s_{ZZ} \end{bmatrix}_a \right] \cdot \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \qquad (22)$$

The collection of the above six observation equations (from Eq. 17 to 22) provides the following single observation equation in matrix form:

$$A \quad X\,\underset{\bullet}{n}\,W \qquad (23)$$

where,

$$A = \begin{bmatrix} -g & g & 0 & 0 & 0 & 0 \\ 0 & 0 & -g & g & 0 & 0 \\ 0 & 0 & 0 & 0 & -g & g \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \qquad (24)$$

$$W = \begin{bmatrix} Av(f_{imu}^{Xup})_X + g & Av(f_{imu}^{Xdn})_X - g & Av(f_{imu}^{Yup})_X & Av(f_{imu}^{Ydn})_X & Av(f_{imu}^{Zup})_X & Av(f_{imu}^{Zdn})_X \\ Av(f_{imu}^{Xup})_Y & Av(f_{imu}^{Xdn})_Y & Av(f_{imu}^{Yup})_Y + g & Av(f_{imu}^{Ydn})_Y - g & Av(f_{imu}^{Zup})_Y & Av(f_{imu}^{Zdn})_Y \\ Av(f_{imu}^{Xup})_Z & Av(f_{imu}^{Xdn})_Z & Av(f_{imu}^{Yup})_Z & Av(f_{imu}^{Ydn})_Z & Av(f_{imu}^{Zup})_Z + g & Av(f_{imu}^{Zdn})_Z - g \end{bmatrix} \quad \ldots(25)$$

$$X = \begin{bmatrix} s_{XX} & s_{XY} & s_{XZ} & b_X \\ s_{YX} & s_{YY} & s_{YZ} & b_Y \\ s_{ZX} & s_{ZY} & s_{ZZ} & b_Z \end{bmatrix}_a, \qquad (26)$$

Now we estimate the calibration parameters as follows:

$$\hat{X} = (W \cdot P.A^T) \cdot (A \cdot P.A^T)^{-1}, \qquad (27)$$

where

$$P = \sigma_0^2 \cdot \Sigma^{-1} \qquad (28)$$

is the 6×6 weight matrix, σ_0^2 is the *a-priori* variance factor (usually $\sigma_0^2 = 1$), and Σ is the sample variance-covariance matrix comprising the sample variances of the accelerometer averages from the six-position test in the diagonal and zeros in the non-diagonal elements. The gyro six-position test with least squares estimation follows the same methodology but the 18 average accelerometer measurements $Av(f_{imu}^{\bullet})$ are replaced by the 18 average gyro measurements $Av(\omega_{imu}^{\bullet})$ and the local gravity g is replaced by the known rotational rate ω_{known} from the turntable.

It should be noted that the static test of high grade inertial sensors can be used to find the scale error of the gyros because the spin of the Earth ($\omega_{Earth} \approx 15.0141 \deg/h$) can be measured. In this case, $\omega_{known} = \omega_{Earth} \cdot \sin(\phi)$ in Eq. (14), where ϕ is the latitude of the inertial sensor position during the calibration test. This case is not valid in the low cost inertial sensors because the Earth's spin is completely buried in high level white noise (low signal to noise ratio). Also, it is worth noting that for low cost inertial sensors, the bias and scale errors are temperature-dependent as indicated by Abdel-Hamid et al. (2004). Therefore, it is strongly recommended to perform the calibration test at different temperature points to estimate the inertial sensor bias and scale errors as functions of temperature.

2.3 Ongoing Research in Calibration

An effective calibration method is the multi-position approach (Shin and El-Sheimy, 2002), based on multiple independent positions of the sensors (18 different positions). This method does not require precise alignment of the IMU axes and can be applied on-the-fly in the field. This method uses the combined three-axis effect of the local gravity and Earth rotational rate to generate the gyro rotational rate excitation signal needed for the calibration. The main disadvantage of this method is that the employed gyro rotation rate excitation signal is the Earth rotational rate, which is a weak signal and can result in observability problems when estimating the scale and non-orthogonality errors. The scale and non-orthogonality errors of low-cost sensors, if not accurately estimated, can contribute significantly to the overall position error during prediction periods (INS-only solutions when GPS outages exist). Thus, instead of using the Earth rotational rate as an excitation signal, Syed et al. (2007) modifies the multi-position calibration method using a rotational rate excitation from a turntable with 26 independent sensor positions (as opposed to the 18 positions in the Earth rotation method). Another advantage of the modified multi-position calibration method is that the least squares singularity problem is resolved efficiently by providing an accurate initial value for the inertial calibration parameters (for more details see Syed et al. (2007)).

In the next section we discuss the different stochastic processes used to model the three residual biases, scale errors and white noise of the gyros and accelerometers (δb_g δb_a, δS_g, δS_a, w_g and w_a).

3. Stochastic Modelling of Inertial Sensor Errors

Various stochastic processes are well documented in Gelb (1974) and Priestley (1981), and their application in inertial navigation is well documented in Jekeli (2000), Grewal et al. (2007) and Rogers (2003). Also, El-Diasty et al. (2007b), Nassar (2005), Flenniken et al. (2005), and Wall and Bevly (2006) are key papers that describe the practical implementation for these stochastic processes and show ongoing research in the area of inertial navigation. The following terms should be defined first (Gelb, 1974; Priestley, 1981):

- Continuous time signals are signals that are described by an analytical function of time.
- Discrete time signals are signals that have values only at discrete instants of time. Sampling a continuous-time signal generates a discrete signal.
- Stationary stochastic process is a process whose joint probability distribution does not change when shifted in time or space. Consequently, parameters such as the mean and variance, if they exist, also do not change over time or with position.
- Autocorrelation function of a discrete signal is the expected value of the product of a random signal with a time-shifted version of itself. If x(t) is random signal then the autocorrelation equation is $R(\tau) = E(x(t) \cdot x(t+\tau))$, where E is the expectation operator and τ is the time shift. The autocorrelation function is very useful because it tells us the time interval over which a correlation in the noise exists.

As mentioned earlier, four stochastic models are described in this note, namely:

- Random constant model,
- Random walk model,
- Gauss-Markov model,
- Autoregressive model

In addition the Allan variance analysis and ongoing research on stochastic modelling are discussed in this note.

3.1 Random Constant (RC) Model

A random bias can be described as an unpredictable random quantity with a constant value through the following differential equation in continuous time domain (Jekeli, 2000):

$$\dot{x} = 0 . \qquad (29)$$

In discrete time, the process is represented by the following equation:

$$x_k = x_{k-1} . \qquad (30)$$

The corresponding autocorrelation function $R(\tau)$ is plotted as a function of time shift τ in Fig. 2 (Gelb, 1974):

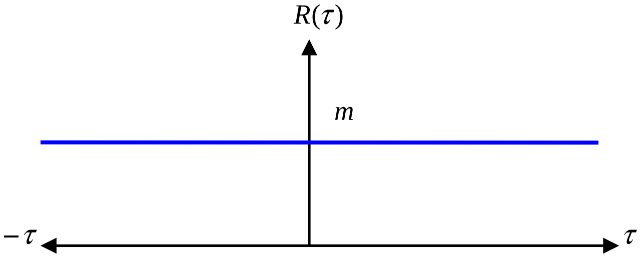


Fig. 2 Autocorrelation function of a random constant process.

Therefore, the corresponding variance is:

$$\sigma_{\hat{x}}^2 = m^2 , \qquad (31)$$

where m is a constant value. So, the discrete-time random constant model can take the form of Eq. (30) but there is no noise present in this process.

3.2 Random Walk (RW) Model

A Random Walk (RW) process x is a zero-mean Gaussian stochastic process with stationary independent increments i.e., in a RW process the difference $(x_k - x_{k-1})$ is a purely random sequence w_k. A RW can be described through the following differential equation in continuous time domain (Jekeli, 2000):

$$\dot{x} = w . \qquad (32)$$

From this equation, it can be seen that RW can be generated by integrating an uncorrelated random sequence *w*. In discrete time, the process can be described through the following equation (Grewal et al., 2007):

$$x_k = x_{k-1} + w_k \,. \qquad (33)$$

For a very large number of data samples k, the previous equation converges to:

$$x_k = \sum_{i=1}^{k-1} w_i \,, \qquad (34)$$

where the mean equals zero and the variance can be derived using the discrete form as follows:

$$\sigma^2_{x_k} = E\left[x_k^2\right] - \mu^2_{\hat{x}} = \sum_{i=1}^{k} E\left[w_i^2\right] = k\sigma^2_w \,, \qquad (35)$$

$$\sigma^2_{w_k} = \frac{\sigma^2_{\hat{x}_k}}{k} \,, \qquad (36)$$

where E is the expectation operator. So, the discrete-time RW model can take the form of Eq. (33) and the variance of the driven noise w_k as Eq. (36). Also, Allan variance analysis can be used to estimate the variance of the driven noise w_k (see section 3.2 for Allan variance details).

3.3 Gauss-Markov Model (Shaping filter)

Gauss-Markov (GM) random processes are stationary processes that have exponential autocorrelation functions. The GM process is important because it is able to represent a large number of physical processes with reasonable accuracy and has a relatively simple mathematical formulation (Gelb, 1974). A stationary Gaussian process that has an exponentially decaying autocorrelation is called first-order GM process. For a random process *x* with zero mean, mean squared error σ^2, and correlation time T_c, the first-order GM model is described by the following continuous-time equation (Gelb, 1974):

$$\dot{x} = -\frac{1}{T_c}x + w \qquad (37)$$

The autocorrelation function (see Fig. 3) of the first-order GM model is given by (Gelb, 1974):

$$R(\tau) = E[x(t)x(t+\tau)] = \sigma^2 e^{-|\tau|/T_C} \qquad (38)$$

where τ is the time shift, T_c is the correlation time, and σ^2 is the variance at zero time shift ($\tau = 0$). The most important characteristic of the GM process is that it can represent bounded uncertainty which means that any correlation coefficient at any time shift is less or equal the correlation coefficient at zero time shift $R(\tau) \le R(0)$) (Gelb, 1974).

Two parameters namely, T_c (correlation time) and σ^2_w (driven noise variance), are required to describe a GM process as shown in Fig. 3.

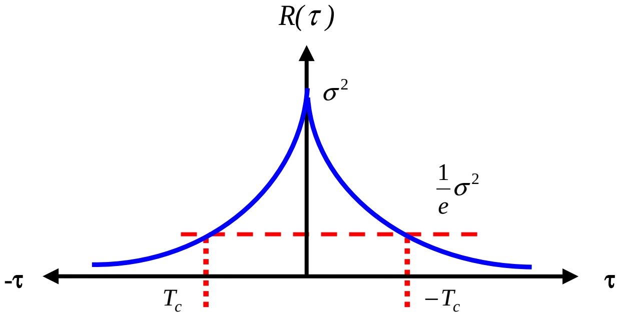


Fig. 3 The autocorrelation function of the first-order Gauss-Markov process.

The first-order GM process in discrete time can be written as (Grewal et al., 2007):

$$x_k = e^{-\Delta t/T_c} x_{k-1} + w_k \,. \qquad (39)$$

And the associated variance can be given by (Grewal et al., 2007):

$$\sigma^2_{x_k} = \frac{\sigma^2_{w_k}}{1 - e^{-2\Delta t_k/T_C}} \,, \qquad (40)$$

$$\sigma^2_{w_k} = \sigma^2_{x_k}\left(1 - e^{-2\Delta t_k/T_C}\right). \qquad (41)$$

So, the discrete-time first-order GM model can be applied using Eq. (39) and the variance of the driven noise w_k is given by Eq. (41).

The second-order GM process with zero mean, mean-square error σ^2, and correlation time T_c, is described by the following continuous-time equation (Gelb, 1974):

$$\ddot{X} = -2\beta \cdot \dot{X} - \beta^2 \cdot X + w \qquad (42)$$

where

$$\beta \approx \frac{2.1416}{T_c} \,. \qquad (43)$$

The autocorrelation function of the second-order GM model is given by (Gelb, 1974):

$$R(\tau)=E[x(t)x(t+\tau)]=\sigma^2(1+\beta\cdot|\tau|)\cdot e^{-\beta\cdot|\tau|} \quad .\ldots(44)$$

An important property of the second-order GM process is that the first derivative of its autocorrelation function equals zero at $\tau=0$. So, we can solve this equation to find the value of β and then we can estimate the correlation time T_c. For higher order GM-processes see Gelb (1974) for more details.

The first-order GM process is one of the most commonly-applied shaping filters in integrated navigation systems because the bounded uncertainty characteristic of GM process makes it the best model for slowly varying sensor errors such as residual bias and scale errors (Rogers, 2003).

3.4 Autoregressive Model

To avoid the problem of inaccurate modelling of inertial sensor random errors due to inaccurate autocorrelation function determination, we can apply another method for estimating inertial sensor errors as introduced by Nassar (2005). Compared to a first-order GM random process, Autoregressive (AR) processes have more modelling flexibilities since they are not always restricted to only one parameter, and higher orders can be used (Nassar, 2005). In many time series applications, AR processes are used to model (estimate) their stochastic part (Gelb, 1974). The inertial sensor data are considered to form a time series that contain both systematic and stochastic error components, and hence AR models are used to describe the inertial stochastic errors. The GM process given by Eq. (37) is equivalent to an AR process of first-order (Nassar, 2005; El-Diasty et al. 2007b). An AR process is a time series produced by a linear combination of past values and its structure is shown in Fig. 4.

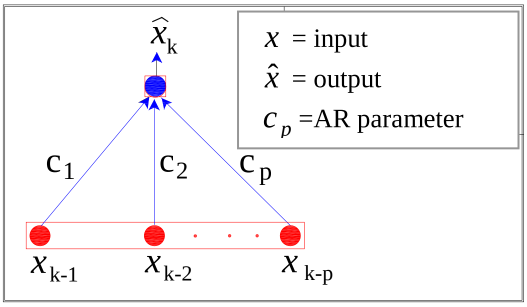


Fig. 4 Autoregressive (AR) structure

An AR process of order p can be described by the following linear equation (Priestley, 1981; El-Diasty et al., 2007a):

$$x_k = \sum_{i=1}^{p} c_i x_{k-i} + w_k , \qquad (45)$$

where x_k is the process output, x_{k-i} are previous system states, and c_i are the AR model parameters. The AR model parameters can be estimated using least-squares fitting (El-Diasty et al., 2007b) or can alternatively be estimated using Yule-Walker, covariance and Burg's methods (Nassar, 2005). The variance of the noise component w_k (is also equivalent to the mean square error MSE in this case because the expected mean of the residual equals zero) can be estimated numerically from the following equation:

$$\sigma^2_{w_k} = \frac{1}{k}\sum_{i=1}^{k}(x_i^d - \hat{x}_i)^2 , \qquad (46)$$

where *k* is the size of the sample of the stationary process, x_k^d is the known value of the process (desired output), and $\hat{x}_k$ is the corresponding estimated output.

If we have a first order AR model, then the discrete form will be (Priestley, 1981):

$$x_k = c_1 \cdot x_{k-1} + w_k \qquad (47)$$

for which the associated variance of the noise component w_k can numerically be estimated from stationary data by Eq. (46) or it can be estimated by using the following equation (Priestley, 1981):

$$\sigma_{w_k} = \begin{cases} \sigma_{x_k}\left(\dfrac{1-c_1^2}{1-c_1^{2k}}\right) & \text{if } |c_1| \neq 1 \\ \dfrac{\sigma_{x_k}}{k} & \text{if } |c_1| = 1 \end{cases} \qquad (48)$$

So, the discrete-time first-order AR model can take the form of Eq. (47) and the variance of the driven noise w_k is given by Eq. (46) or Eq. (48). It should be noted that when c_1=1, the AR process becomes a RW process. The AR model was introduced by Nassar (2005) as an alternative to GM process for the modelling of the residual gyro and accelerometer biases.

3.5 Allan Variance Analysis

Allan variance analysis is commonly and efficiently used to identify and obtain the variances for most of the random errors (IEEE Std. 647-1995, 1998; Hou and El-Sheimy, 2003; El-Diasty et al., 2007a). The Allan variance is a method of representing root mean square random drift error as a function of averaging times. It is simple to compute, much better than having a single RMS drift number to apply to a system error analysis, and relatively simple to interpret and understand. Its most useful application is in the identification and estimation of random drift coefficient in a previously formulated model equation. If N is the number of data points with sampling internal of Δt_0, then a group of n data points (with $n < N/2$ can be created. Each group member is called a cluster T of size $n\Delta t_0$. The Allan variance can be defined in terms of an output variable, calculated at discrete times $x_k = x(kt_0)$. The Allan variance is estimated as follows:

$$\sigma^2(T) = \frac{1}{2T^2(N-2n)} \sum_{k=1}^{N-2n} (x_{k+2n} - 2x_{k+n} + x_k)^2 \quad \ldots\ (49)$$

There is a very important relationship between Allan variance and power spectral density (PSD) of a random process:

$$\sigma^2(T) = 4\int_0^{\infty} df \cdot S_x(f) \cdot \frac{\sin^4(\pi f T)}{(\pi f T)^2} \qquad (50)$$

where $S_x(f)$ is the power spectral density (PSD) of the random process $x(T)$, namely the instantaneous output rate of the sensor. In the derivation of Eq. (50), it is assumed that the random process $x(T)$ is stationary.

Eq. (50) is the equation that will be used to calculate the Allan variance from the PSD. The different types of random processes can be examined by investigating the Allan variance plot. The Allan variance provides a means of identifying various noise terms that exist in the data. Fig. 5 shows a typical Allan variance curve estimated from gyro measurements. A typical Allan variance curve estimated from accelerometer measurements is the same as from a gyro but the angle random walk and the rate random walk terms should be changed to velocity random walk and the acceleration random walk terms in the plot. There are four possible RW models in inertial navigation systems. The *angle RW* that describes the angular error as a function of time is due to the mathematical integration of the white noise (w_g) of the angular rate (gyro output). However, the residual bias (b_g) of the gyro can be modelled as rate RW process. On the other hand, the velocity error as a function of time that is due to the mathematical integration of white noise (w_a) of the linear acceleration (accelerometer output) is called velocity RW and the residual bias (b_a) of the accelerometer can be modelled as *acceleration RW* process. The Allan variance terms and algorithm are well documented in IEEE standards (1998)

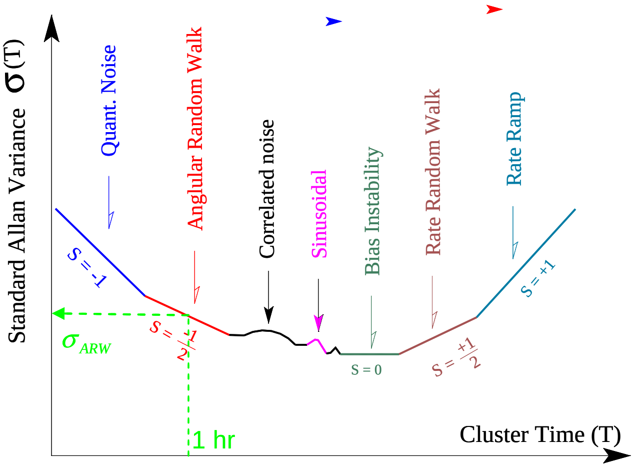


Fig. 5 $\sigma(T)$ Allan variance analysis noise terms results (after IEEE Std. 952-1995, 1998)

It should be noted that different noise terms appear in different regions of T. This property permits easy identification of various random processes that exist in the data. It is well known that the angular and velocity random walk are the dominant noise terms in low cost inertial sensors and therefore we provide in this note how the angle RW (in case of gyros) or velocity RW (in case of accelerometer) can be identified from the Allan plots. The angular random walk process can be identified at T=1h and with a straight line of slope -1/2 as shown in Fig. 5. The noise PSD rate is represented by (IEEE Std. 647-1995, 1998):

$$S_x(f) = \sigma_{ARW}^2, \qquad (51)$$

where σ_{ARW} is the angular random walk coefficient from Fig. 5. Substituting Eq. (51) in Eq. (50) and performing the integration we get:

$$\sigma^2(T) = \frac{\sigma_{ARW}^2}{T}. \qquad (52)$$

The same estimation can be made to find the velocity random walk from process from the typical Allan variance curve of the accelerometer. The identification of the remaining various random processes that exist in the data can easily be derived using the same methodology given that the slope of Allan variance is well known for any individual random process under investigation as shown in Fig. 5.

3.6 Ongoing Research on Stochastic Modelling

The most commonly used process in stochastic modelling of inertial sensor errors is the first-order Gauss-Markov process, while recently, the use of Autoregressive (AR) modelling methods were tested (Nassar, 2005; Park and Gao, 2002). Nassar (2005) implemented the modelling of the inertial sensor gyro and accelerometer residual biases (three gyros and three accelerometers) using AR processes of different orders and showed that the accuracy of position is improved by almost 50% for second-order AR model and 55% for third-order AR model when compared with the first-order GM and AR model results (Nassar, 2005). However, the number of INS sensor error states is increased from six states to 6×2 for second order AR model or 6×3 for third order AR model. In addition, the implementation of the Kalman filter with large number of states becomes numerically intense and complicates the model excessively (see Nassar, 2005 for more details).

Most recently El-Diasty et al. (2007b) proposed nonlinear stochastic model using wavelet networks. They introduced a new nonlinear stochastic model for inertial sensor residual biases and verified its performance in comparison with first order GM and AR. It was found that the wavelet network-based nonlinear stochastic process can be used to model the highly nonlinear time-varying inertial sensor error. A kinematic test with nine artificial GPS outages of 30s and 60s each showed that the first-order GM and AR stochastic processes give similar results, which agree with the results obtained by Nassar (2005). In addition, the first-order WN-based nonlinear stochastic model gives superior results to the first-order GM and AR processes with an overall improvement of 30% in the 3D position solution for 30s and 60s GPS outages (see El-Diasty et al., 2007b for more details). To this end, in the next section we discuss the implementation of a stochastic model in the INS/GPS navigation filter.

Also, it is worth noting that for low cost inertial sensors, the residual biases are temperature-dependent as indicated by El-Diasty et al. (2007a). Therefore, it is strongly recommended to develop the stochastic model for residual biases at different-temperature points.

4. Stochastic Model Implementation in Loosely-coupled INS/GPS Integration

Methods in which GPS and INS data are integrated differ mostly in the type of data that are shared between the systems. In general however, the following four approaches are the most common (Jekeli, 2000): uncoupled integration, loosely-coupled integration, tight integration, and deep integration. The loosely-coupled and tightly-coupled integration strategies are the most common in practice. In this note we discuss the implementation of the stochastic models in a loosely-coupled integration scheme. In the loosely-coupled integration strategy, position and velocity are used as observations to an INS-only filter. The position and velocity estimates are obtained from a GPS-only filter. This way, the integration approach uses a cascading scheme in which the raw GPS measurements are first processed in a GPS-only filter before they get passed along to aid the INS-only filter (Jekeli, 2000). The inertial navigation error state behaviour is obtained by the perturbation of the INS mechanization equations. This perturbation analysis is well documented in a number of publications, such as Jekeli (2000), Titterton (2004), and Grewal et al. (2007). The error model comprising errors in INS navigation states (i.e., three residual positions $\delta \mathrm{p}$, three residual velocities $\delta \mathrm{v}$, and three residual attitudes in Euler angles $\delta \mathrm{A}$) as well as the INS sensor errors (i.e., three gyro residual biases $\delta \mathrm{b}_{\mathrm{g}}$, three residual scale errors $\delta \mathrm{s}_{\mathrm{g}}$, three accelerometer residual biases $\delta \mathrm{b}_{\mathrm{a}}$, and three residual scale errors $\delta \mathrm{s}_{\mathrm{a}}$) are used. The system of discrete linearized first-order differential equations for inertial system error model and GPS measurements is used to provide complete navigation solution (positions, velocities and attitude) using INS/GPS integration in standard loosely-coupled mode. The state vector for loosely-coupled INS/GPS error model can be represented by (if we have 21 states):

$$\mathbf{x}_{\mathrm{k}} = \Phi_{\mathrm{k}-1} \cdot \mathbf{x}_{\mathrm{k}-1} + \mathbf{w}_{\mathrm{k}} , \tag{53}$$

$$\text{where, } \mathbf{x} = [\delta \mathrm{p}_{(1\times3)}, \delta \mathrm{v}_{(1\times3)}, \delta \mathrm{A}_{(1\times3)}, \Big| \quad \delta \mathrm{b}_{\mathrm{g}(1\times3)}, \delta \mathrm{b}_{\mathrm{a}(1\times3)}, \delta \mathrm{s}_{\mathrm{g}(1\times3)}, \delta \mathrm{s}_{\mathrm{a}(1\times3)}]^{\mathrm{T}},$$

which contains two parts separated by vertical line: The first part is called the inertial dynamic model which contains the three position, velocity and attitude errors ($\delta \mathrm{p}_{(1\times3)}, \delta \mathrm{v}_{(1\times3)}$ and $\delta \mathrm{A}_{(1\times3)}$), which are derived from the perturbation of the INS mechanization equations (see Jekeli, 2000 for this perturbation analysis). The second part is called the stochastic model, which contains the three gyro and accelerometer residual biases and scale errors ($\delta \mathrm{b}_{\mathrm{g}(1\times3)}, \delta \mathrm{b}_{\mathrm{a}(1\times3)}, \delta \mathrm{s}_{\mathrm{g}(1\times3)}$ and $\delta s_{a(1\times3)}$). $\Phi_{\mathrm{k}-1}$ is the transition matrix which contains the parameters from the dynamic and stochastic model, and

$$w_k = [w_{\delta v(1\times3)}, w_{\delta A(1\times3)}, w_{bg_{(1\times3)}}, w_{ba(1\times3)}, w_{Sg_{(1\times3)}}, w_{Sa(1\times3)}]^T$$

is the vector comprising the noise components which follow the standard normal distribution $w_k \sim N(0, Q_k)$ where Q_k is the covariance matrix with a diagonal form and includes the following variances in a discrete form:

$$Q_k = \mathrm{diagonal}[\sigma^2_{w\delta v(1\times3)}, \sigma^2_{w\delta A(1\times3)}, \sigma^2_{wbg_{(1\times3)}}, \sigma^2_{wba(1\times3)}, \sigma^2_{wsg_{(1\times3)}}, \sigma^2_{wsa(1\times3)}]^T.$$

The correct identification of the stochastic processes determined above for specific inertial sensors is of major importance in the performance of an integrated model and especially in the ability of a pure inertial solution to bridge data outages. In this note we give an example of how we can extract the stochastic model parameters from the specification sheet of Digital Quartz Inertial (DQI-100) sensor from BEI Systron Donner Inertial Division (BEI, 2004). The DQI is a low cost tactical grade inertial measurement unit based on quartz gyro and accelerometer technology (BEI, 2004).

The first set of parameters to be retrieved comprises the uncertainties ($\sigma_{w\delta v(1\times3)}$ and $\sigma_{w\delta A(1\times3)}$) of the three velocity and attitude error model, respectively. The processes in this case are three angle random walk (ARW) and three velocity random walk (VRW). For simplicity, the ARW and VRW variances are obtained from the specification sheet of DQI-100 (BEI, 2004). Usually, the power densities of ARW and VRW are given in the specification sheet and when we have a dynamic system in the discrete form the associated variances can be estimated as follows:

(1) From DQI-100 specification sheet we know that $\mathrm{ARW} = 0.035\,\mathrm{deg}/\sqrt{\mathrm{h}} = 2.10\ \ \mathrm{deg}/\mathrm{h}/\sqrt{\mathrm{Hz}}$, then for 100 Hz bandwidth (sampling rate of 0.01 sample/sec), the attitude error noise uncertainty equal:

$$\sigma_{w\delta A} = 2.1\mathrm{deg}/\mathrm{h}/\sqrt{\mathrm{Hz}} \times \sqrt{100\mathrm{hz}} = 21\mathrm{deg}/\mathrm{h}\ .$$

(2) Again from DQI-100 specification sheet we know that $\mathrm{VRW} = 60\,\mu g/\sqrt{\mathrm{Hz}}$ (where g is the local gravity), then the velocity noise uncertainty equal:

$$\sigma_{w\delta v} = 60\mu g/\sqrt{\mathrm{Hz}} \times \sqrt{100\mathrm{Hz}} = 600\mu g\,.$$

Now, we investigate the three gyro and accelerometer residual biases ($\delta b_{g_{(1\times3)}}$ and $\delta b_{a(1\times3)}$). Traditionally, the residual biases are modelled as Gauss-Markov (GM) processes (Rogers, 2003). The autocorrelation function of the stationary raw data is used to determine the parameters of the Gauss-Markov models. It must be noted however that, prior to the calculation of the autocorrelation function, the inertial sensor data should be de-noised using a low pass filter or wavelet or neural network de-noising. The parameters are derived for each sensor individually. Table 1 shows the correlation time T_b (subscript *b* means bias) and uncertainty for the residual bias errors from the DQI-100 inertial unit specification sheet (BEI, 2004). Noise uncertainty and correlation time are illustrated in the table.

Table 1 First order GM process parameters

Stochastic model	**Correlation Time T_b**	**Uncertainty σ**
Gyros residual bias δb_g	60 s	3 deg/h
Acc residual bias δb_a	60 s	$200\mu g$

If we consider that the sampling interval is Δt_{k-1} then the GM model for the residual biases can be written as follows:

$$\delta b_k = e^{-\Delta t_{k-1}/60}\delta b_{k-1} + w_{k-1}\,. \qquad (54)$$

We should note that the sampling interval Δt_{k-1} is not exactly constant due to the INS/GPS acquisition and synchronization issues.

Finally, we investigate the three residual scale errors for gyros and accelerometers ($\delta s_{g_{(1\times3)}}$ and $\delta s_{a(1\times3)}$). The scale error is mostly deterministic in nature and only suffers a small residual error due to temperature variation and nonlinearity. It is impossible from the practical point of view, and in static mode, to differentiate the effect of residual scale error and residual bias terms. As such, the residual scale error is modelled as a random constant (RC) (Cannon, 1991). An alternative approach is to model the residual scale error using GM process (Rogers, 2003). In this case, the correlation time T_s (subscript *s* means scale) and the noise uncertainty will be tuned in the navigation filter to provide the best estimation. It should be noted that the navigation solution in this case will be sub-optimal because the residual scale error parameters used are based on the tuning method and not on a rigorous min/max method.

In summary, if we model the residual bias, and scale errors as GM processes, the stochastic model takes the following form based on the example given in this section:

$$\begin{bmatrix} \delta b_g^x \\ \delta b_g^y \\ \delta b_g^z \\ \delta b_a^x \\ \delta b_a^y \\ \delta b_a^z \\ \delta s_g^x \\ \delta s_g^y \\ \delta s_g^z \\ \delta s_a^x \\ \delta s_a^y \\ \delta s_a^z \end{bmatrix}_k = \begin{bmatrix} c_{bg}^x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & c_{bg}^y & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & c_{bg}^z & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{ba}^x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{ba}^y & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & c_{ba}^z & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & c_{sg}^x & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{sg}^y & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{sg}^z & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{sa}^x & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{sa}^y & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{sa}^z \end{bmatrix} \cdot \begin{bmatrix} \delta b_g^x \\ \delta b_g^y \\ \delta b_g^z \\ \delta b_a^x \\ \delta b_a^y \\ \delta b_a^z \\ \delta s_g^x \\ \delta s_g^y \\ \delta s_g^z \\ \delta s_a^x \\ \delta s_a^y \\ \delta s_a^z \end{bmatrix}_{k-1} + \begin{bmatrix} w_{bg}^x \\ w_{bg}^y \\ w_{bg}^z \\ w_{ba}^x \\ w_{ba}^y \\ w_{ba}^z \\ w_{sg}^x \\ w_{sg}^y \\ w_{sg}^z \\ w_{sa}^x \\ w_{sa}^y \\ w_{sa}^z \end{bmatrix}_k \qquad \ldots (55)$$

where $c_{bg}^x = c_{bg}^y = c_{bg}^z = c_{ba}^x = c_{ba}^y = c_{ba}^z = e^{-\Delta t_{k-1}/60}$, the residual bias error correlation time equals 60s as shown in Table 1 and $c_{sg}^x = c_{sg}^y = c_{sg}^z = c_{sa}^x = c_{sa}^y = c_{sa}^z = e^{-\Delta t_{k-1}/T_s}$ where the residual scale error correlation T_s is tuned (as mentioned before) to the navigation filter to provide the best estimation for positions, velocities and attitudes. The example given here is based on the correlation time being the same for all residual bias errors because we simply used the correlation time from the inertial sensor specification sheet. However, in practice we use static test to collect the three gyro and three accelerometer measurements from the inertial sensors. From the autocorrelation sequence we can estimate three different correlation times for the three gyros residual errors and three different correlation times for the three accelerometers residual biases.

Acknowledgments

This technical note was supported partially by the Natural Sciences and Engineering Research Council (NSERC) of Canada and by the GEOIDE National Centre of Excellence (Canada). Also, revision of this technical note by anonymous reviewers and discussions with Dr. Jian-Guo Wang, are greatly appreciated by the authors.

References

Abdel-Hamid, W., Osman, A., Noureldin, A. and El-Sheimy, N. (2004) "***Improving the Performance of MEMS-based Inertial Sensors by Removing Short-Term Errors Utilizing Wavelet Multi-Resolution Analysis.***" Proceedings of the ION NTM, San Diego, CA.

BEI Technologies, Inc. (2004) "***C-MIGITSTM III User's Guide.***" Concord, California 94518-1399, USA.

Cannon, M.E. (1991) "***Airborne GPS/INS with an Application to Aerotriangulation.***" Ph.D. thesis, University of Calgary, UCGE Report 20040.

El-Rabbany, A. and El-Diasty, M. (2004) "***An efficient neural network model for de-noising of MEMS-based inertial data.***" The Journal of Navigation, 57: 407-415.

El-Diasty, M., A. El-Rabbany, A. and S. Pagiatakis (2007a) "***Temperature Variation Effects on Stochastic Characteristics for Low Cost MEMS-based Inertial Sensor Error.***" Measurement Science and Technology, Vol. 18, No.11, pp. 3321-3328.

El-Diasty, M., A. El-Rabbany, A. and S. Pagiatakis (2007b) ***"An Accurate Nonlinear Stochastic Model for MEMS-Based Inertial Sensor Error with Wavelet Networks."*** Journal of Applied Geodesy Vol. 1, No. 4, pp.201-212.

IEEE Std. 647-1995 (1998) **"*IEEE Standard Specification Format Guide and Test Procedure for Single-axis Interferometric Optic Gyros.*"**

Flenniken, W., Wall, J., Bevly, D.M. (2005) **"*Characterization of Various IMU Error Sources and the Effect on Navigation Performance.*"** Proceedings of ION GNSS, Long Beach, CA.

Gelb, A. (1974) **"*Applied Optimal Estimation.*"** The M.I.T. Press, Cambridge, Massachusetts.

Grewal, M, L. Weill and A. Andrews (2007) **"*Global Positioning Systems, Inertial Navigation, and Integration* ."** John Wiley & Sons, Inc.

Hou H and El-Sheimy N (2003) **"*Inertial sensors errors modeling using Allan variance.*"** Proceedings of ION GPS, Portland OR.

Jekeli, C. (2000) **"*Inertial Navigation Systems with Geodetic Applications.*"** Walter de Gruyter, New York, NY., USA.

Nassar, S. (2005) **"*Accurate INS/DGPS positioning using INS data de-noising and autoregressive (AR) modeling of inertial sensor errors.*"** Geomatica, 59(3), 283-294.

Park, M. and Gao, Y. (2002) **"*Error Analysis of Low-Cost MEMS-based Accelerometers for Land Vehicle Navigation.*"** Proceedings of ION GPS, Portland, Oregon.

Priestley, M.B. (1981) **"*Spectral Analysis and Time Series.*"** Vol. 1, Academic Press.

Rogers, Robert M. (2003) **"*Applied mathematics in integrated navigation systems.*"** 2nd ed. AIAA education series.

Skaloud, J (1999) **"*Optimizing Georeferencing of Airborne Survey Systems by INS/DGPS.*"** Ph.D. Thesis, Dept. of Geomatics Eng., The University of Calgary, Calgary, Alberta, Canada.

Salychev, O. (1998) **"*Inertial Systems in Navigation and Geophysics.*"** Bauman MSTU Press, Moscow.

Shin, E.-H. and El-Sheimy, N. (2002) **"*Accuracy Improvement of Low Cost INS/GPS for Land Applications.*"** Proceedings of ION NTM, San Diego, CA.

Syed Z. F., Aggarwal P., Goodall C., Niu X. and El-Sheimy N. (2007) **"*A new multi-position calibration method for MEMS inertial navigation systems.*"** Measurements Science and Technology, No. 18 (2007), pp. 1897-1907.

Titterton, D. H. (2004) **"*Strapdown inertial navigation technology.*"** 2nd ed., Radar, sonar, navigations & avionics.

Wall, J. and Bevly, D. M. (2006) **"*Characterization of Inertial Sensor Measurements for Navigation Performance Analysis.*"** Proceedings of the ION GNSS, Fort Worth, TX.